# Robotic self-exploration and acquisition of sensorimotor skills

Dissertation

zur Erlangung des akademischen Grades
doctor rerum naturalium (Dr. rer. nat.)

im Promotionsfach Informatik

eingereicht an der

Mathematisch-Naturwissenschaftlichen Fakultät
der Humboldt-Universität zu Berlin

von
Dipl. Inf. Oswald Berthold

Präsidentin der Humboldt-Universität zu Berlin:
Prof. Dr.-Ing. habil. Dr. Sabine Kunst

Dekan der Mathematisch-Naturwissenschaftlichen Fakultät:
Prof. Dr. Elmar Kulke

Gutachter*innen:

1. Prof. Dr. Verena V. Hafner (HU, Berlin)
2. Prof. Dr. Björn Scheuermann (HU, Berlin)
3. Dr. Georg Martius (Max Planck Institut, Tübingen)

Eingereicht am: 12. Juni 2018
Verteidigt am: 17. Juni 2019

# Abstract

The interaction of machines with their environment should be reliable, safe, and ecologically adequate. To ensure this over long-term complex scenarios, a theory of adaptive behavior is needed. In developmental robotics and embodied artificial intelligence, among other fields of study, behavior is approached as a phenomenon that emerges from an ongoing dynamic interaction between entities called agent, body, and environment. In this context, the theory should allow to make quantitative predictions about the behavior knowing a particular configuration, and inversely it should be able to make predictions about required functions of agents and the body to cope with a particular environment.

This thesis investigates generative models of adaptive behavior on *robots* that are able to learn *rapidly* and *on their own*, how to do *primitive motions*, using only sensorimotor information. The aim in the long-run is to reuse acquired skills when learning other motions in the future, and thereby *grow* a complex repertoire of possible interactions with the world, that is *fully grounded* in, and *continually adapted* to sensorimotor experience through developmental processes.

Using methods from machine learning, computational neuroscience, statistics, and physics, the question is decomposed into the relationship of representation, exploration, and learning. A framework is provided for systematic variation and evaluation of models, using a set of different measures. The proposed framework considers procedural generation of hypotheses as scientific workflows using a fixed set of functional building blocks, and allowing to search for models by seamless evaluation in simulation and real world experiments. The *self* in the title of the thesis has double meaning, referring both to an autonomous learning drive, as well as to the agent's capacity for self-other distinction, both considered functional requirements of complex adaptive systems.

Additional contributions of the thesis are related to the agent's causal footprint in sensorimotor time. A probabilistic graphical model is provided, along with an information-theoretic learning algorithm, to discover networks of information flow in sensorimotor data. A generic developmental model, based on real time prediction learning, is presented and discussed on the basis of three different algorithmic variations. The discussion is closed with some perspectives on growth and scaling.

**Keywords**: Sensorimotor skills; Adaptive behavior; Learning and autonomy; Online algorithms;

# Zusammenfassung

**Selbst-exploration und Aneignung von sensomotorischen Fertigkeiten in Robotern**

Die Interaktion zwischen Maschinen und ihrer Umgebung sollte zuverlässig, sicher und ökologisch adequat sein. Um das in komplexen Szenarien langfristig zu gewährleisten, wird eine Theorie adaptiven Verhaltens benötigt. In der Entwicklungsrobotik und verkörperten künstlichen Intelligenz wird Verhalten als emergentes Phänomen auf der fortlaufenden dynamischen Interaktion zwischen Agent, Körper und Umgebung betrachtet. Die gewünschte Theorie sollte quantitative Vorhersagen über Verhalten auf Basis der Konfiguration machen können, genauso wie Vorhersagen über die funktionalen Anforderungen an Agent und Körper auf Grundlage einer Umgebung.
Diese Arbeit untersucht generative Modelle adaptiven Verhaltens von Robotern die in der Lage sind, schnell und selbständig einfache Bewegungen zu erlernen, ausschliesslich auf Grundlage sensomotorischer Information. Das langfristige Ziel dabei ist die Wiederverwendung gelernter Fertigkeiten in späteren Lernprozessen um damit ein komplexes Interaktionsrepertoire mit der Welt entstehen zu lassen, das durch Entwicklungsprozesse vollständig und fortwährend adaptiv in der sensomotorischen Erfahrung verankert ist.
Unter Verwendung von Methoden des maschinellen Lernens, der Neurowissenschaft, Statistik und Physik wird die Frage zerlegt in die Komponenten Repräsentation, Exploration, und Lernen und deren gegenseitige Beziehungen. Es wird ein Gefüge für die systematische Variation und Evaluation von Modellen errichtet unter Verwendung verschiedener Maße. Das vorgeschlagene Rahmenwerk behandelt die prozedurale Erzeugung von Hypothesen als Flussgraphen über einer festen Menge von Funktionsbausteinen, was die Modellsuche durch nahtlose Anbindung über simulierte und physikalische Systeme hinweg ermöglicht.
Ein Schwerpunkt der Arbeit liegt auf dem kausalen Fussabdruck des Agenten in der sensomotorischen Zeit. Dahingehend wird ein probabilistisches graphisches Modell vorgeschlagen um Informationsflussnetzwerke in sensomotorischen Daten zu repräsentieren. Das Modell wird durch einen auf informationtheoretischen Grössen basierenden Lernalgorithmus ergänzt. Es wird ein allgemeines Modell für Entwicklungslernen auf Basis von Vorhersagelernen in Echtzeit präsentiert und anhand von drei Variationen näher besprochen. Die Darstellung endet mit der Betrachtung von Wachstum und Skalierung.

**Schlagworte**: Sensomotorische Fertigkeit; Adaptives Verhalten; Lernen und Autonomie; Echtzeit-Algorithmen;

# Contents

# Part I.

# Introduction

# 1. Motivation

The organization of nervous systems and brains is only partially understood, although a lot of research in different disciplines has led to a sizable amount of interesting results and surprising insights. Artificial intelligence research is characterized by a synthetic methodology, which is also expressed as the law of uphill analysis and downhill invention (Braitenberg 1984, p. 20). Bio-inspired approaches in artificial intelligence search for biological functions and principles with the aim of reproducing them in hardware or algorithms (Floreano and Mattiussi 2008). A lot of recent progress in robotics research has been made based on this approach, with actual products out on the market, for example in the drone and automotive segments. These are direct outcomes of bio-inspired research on insect vision and miniature flying robots. Evolutionary algorithms represent another very active field of research, with broad applications beyond basic research, and proven in the field. Evolutionary methods are widely applied in robotics research, defining an entire subfield (Stefano Nolfi 2000) that is subject to ongoing research (Cully et al. 2015; Doncieux et al. 2015), and producing state of the art results.

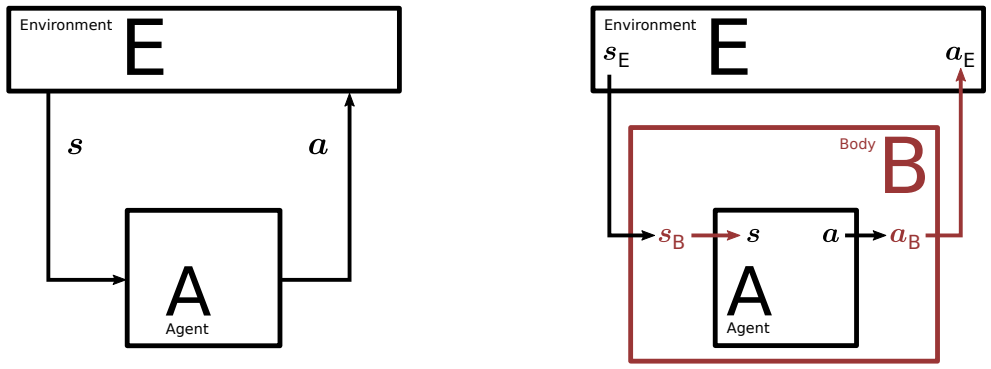
This work is on developmental robotics, which investigates the principles and organization of epigenetic development and learning in humans at various different stages inspired by theories of cognitive development (Weng et al. 2001; Lungarella et al. 2003; M. Asada et al. 2009).

Complications from real world physics are still a challenge for robotics. One answer to this is the *embodiment hypothesis* of intelligent behavior (Pfeifer and Bongard 2007). It has become clear, that the body has huge influence on the complexity of the robots task in terms of its sensorimotor space, and the representation of action and perception with respect to their task relevance. This has been extended to the concept of *morphological computation*, where the body is said to perform useful information processing (Hauser et al. 2011; V. C. Müller and Hoffmann 2017). There are convincing arguments, that real world intelligent behavior cannot be meaningfully understood without considering the body. The addition of the body is reflected in the modified diagram shown in Figure 1.1b.

Evidence from early learning in humans suggests that self-exploration and learning the particulars of the body is a fundamental part of the schedule, and an important precedent for all higher learning ability later on. This is plausible in view of the overall challenge of learning the sensorimotor control task (Bernshteĭn 1967), that developmental approaches try to frame entirely in terms of the interaction of adaptive components. The level of motion skills is relevant for basically any type of robot. The problem examined in this thesis is largely inspired by quadrotor flying robots, which were the subject of previous work (V. V. Hafner et al. 2010; Berthold, M. Müller, and V. V. Hafner 2011).

A core concept in AI is the agent, which is anything that can sense and interact with its outside, defined in (Russell and Norvig 2003) as

> An agent is anything that can be viewed as *perceiving* its environment through *sensors*




(a) Agent - environment

(b) Agent - body - environment

Figure 1.1.: Agent - environment interaction via actions $\boldsymbol{a}$ and states $\boldsymbol{s}$ on the right in Figure 1.1a. Agent - body - environment interaction is shown on the right in Figure 1.1b. The original action $\boldsymbol{a}$ is transformed into the body reference system as $\boldsymbol{a}_\mathrm{B}$, and then into the environmental context as $\boldsymbol{a}_\mathrm{E}$. This happens analogously with the state $\boldsymbol{s}$.

and *acting* upon that environment through *effectors*.

This is illustrated in Figure 1.1a showing the agent A next to an environment E, with interaction taking place via action $\boldsymbol{a}$ and state perception $\boldsymbol{s}$. The agent sends an actions to the environment and receives a state in return. The process of performing an action and observing the resulting state is repeated forever during the lifetime of an agent. The sequence of actions and states produced by the agent in this way is its *behavior*. An agent computes actions as a function of states,

$$\boldsymbol{a} = \mathrm{A}(\boldsymbol{s}) \tag{1.1}$$

This update rule in Equation 1.1 is called a *controller*, a *strategy*, an *inverse model* or a *policy*, depending on the context. This is general enough to allow discussion of artificial and biological agents. A developing robot is an instance of the class of agents that interact with a real physical environment, inherently subject to partial observability.

As a conrete example consider a quadrotor, commonly called a drone. It is a helicopter with four rotors mounted on a rigid frame with many possible variations of the exact configuration. For a human it is generally not possible, to fly a quadrotor using the raw motor signals, thus an onboard autopilot provides angular stabilization using inertial sensors for estimating the robot pose. The robot's linear motion in three dimensions results from its angular configuration. This is a challenging dynamic control task and takes a bit of training for humans to achieve a comfortable level of performance. The first problem is to regulate the total thrust to keep the robot at a fixed height, the second problem is that the robot will drift from its horizontal position due to accumulated errors.

The motivation is to be able to model the process of learning to fly a drone as a human. This in turn requires an understanding of how an agent can explore its own body and its immediate surroundings to acquire control strategies starting with primitive motions, and using only data directly available from the sensorimotor level.

# 2. Problem statement

With the motivation in mind, the problem statement for this thesis can be given as

> How can a robot learn about its body from scratch?

Stabilization and motion of the robot in three dimensions of space constitutes a dynamic control problem that can be solved using adaptive control techniques (Ioannou and Sun 1996; Ng and Kim 2004). Although effective, these techniques are often based on implicit assumptions about the properties of a specific system, that may not hold for more general autonomous development in robots. Here, the problem is examined as learning controllers from completely uninformed priors (from scratch) in a fully embodied setting. The force-controlled nature of rigid body motion in free space motivates the point mass model, which is a highly configurable inert rigid body robot model, and used throughout the experiments. The basic point mass state update equation for a configuration $\mathrm{pm}$ is given by

$$\begin{aligned} \boldsymbol{s}_{t+1} &= h(\boldsymbol{M}\boldsymbol{s}_t + \boldsymbol{u}_{t-\mathsf{lag}}) + \boldsymbol{n}_t && (2.1) \\ \boldsymbol{u}_t &= \mathsf{A}_i(\boldsymbol{s}_t) && (2.2) \end{aligned}$$

with agent $\mathsf{A}_i$, fully described by parameters $i$, $\dim = |\mathsf{A}|$ the state dimension of the agent, state $\boldsymbol{s} \in \mathbb{R}^{\dim}$, element-wise nonlinear transfer function $h$, state update matrix $\boldsymbol{M}$, motor input $\boldsymbol{u}$ and noise term $\boldsymbol{n}$. A more detailed description is given in chapter A of the Appendix. The motor input $\boldsymbol{u}$ is computed by the $\mathsf{A}_i$ function in Equation 2.2 from the current state. Given a measure $P$ on the state $\boldsymbol{s}$, the problem can be solved by descending the gradient of the measure with respect to the parameters $i$,

$$\nabla_i \sum_t P(\boldsymbol{s}_t) \tag{2.3}$$

until some threshold $P_{\mathsf{crit}}$ is met by a moving average $\bar{P}$ of $P$. If the gradient search of Equation 2.3 is evaluated over increasingly different robots, with a selection of those used in this work shown in Figure 2.1, it is natural to ask about systematic variations in the measure $P$ over different bodies and environments with respect to the agent function $\mathsf{A}_i$. To start to answer this question, it is proposed to frame it as a variational problem that can again be approached using a search process

$$\underset{\mathsf{A}_i,(\mathsf{B},\mathsf{E})_j}{\operatorname{argmin}} \sum_t P(\boldsymbol{s}_t) \tag{2.4}$$

Each evaluation $(i, j)$ of duration $k$ produces an episode that is represented by a $k \times \dim$ matrix $\boldsymbol{S}$, which is collected into a data-set of episodes represented by the tensor $\boldsymbol{T}$.
It can then be attempted to mine the data-set for answers in terms of statistical dependencies. Ideally, it should be possible at some point, to predict the requirements for a learning agent, based on what is known about the body and environmental properties, while satisfying criteria of safety and effectiveness. This is certainly beyond the scope of this work, but the methodology of iterative model search (Biswal et al. 2010; J. S. Bergstra et al. 2011; J. Bergstra, Yamins, and Cox 2013; Cully et al. 2015) over random robots sampled from a given population is a guiding principle of this thesis.
Some examples of the beauty of learning processes are "learning to fly like a bird" (Tedrake et al. 2009), learning to ride a bicycle (Cook and Bruck 2004) or rock climbing. In the last case, good climbers are not good because they have learned all problems, but because they have accumulated moves for on-the-fly exploration as in foraging.


Figure 2.1.: The entire population of different quadrotors in the lab in 2014 is shown in the top picture. In this project, the one in the bottom left corner, and that in the top right corner, shown in more detail in the bottom row, were built and used. The initial experiments have been extended to all other platforms shown in the middle and bottom row. This includes commercial ones like the Turtlebot shown in the middle left, the Nao in the middle right, or the Sphero in the bottom right, as well as custom designs such as the hucar in the center of the figure. Quadrotor zoo photo by C. Blum, 2014, used with permission, the Nao photo by Edsiekanschrijven, 2014, Wikimedia Commons, `https://commons.wikimedia.org/wiki/File:Nao_Robot_Close_up.JPG`.

# 3. Approach and definitions

The bio-inspired approach taken in this thesis allows to pose survival as the question of "Where to go and how to get there?". Successful agents are required to include a function for answering this question repeatedly, and in different circumstances.
In this form, the problem and its solutions are referred to as *teleology* (Rosenblueth, Wiener, and Bigelow 1943). A necessary criterion frequently used for assessing the intelligence of a given behaviour is the *pursuit of goals*, and to a large extent, goals exert themselves as spatial problems solved by motion (Otto E. Rössler 1974). The most rudimentary goal directed behaviour needs at least a *goal recognition* mechanism (Franz and Mallot 2000) to modulate an innate behaviour. Modulating the motor activity with the goal recognition signal results in *random search* or *kinesis*. Increasingly sophisticated strategies of goal pursuit can be stacked on top of this baseline. The result is the huge diversity of strategies observed across the biome. This is also desired in complex robot behaviours. It is an open research question, if and to what extent goals are part of the exploration itself. One hypothesis investigated here, is that goals are not special in their *goalness* but are bottom-down predictions themselves, that is, goals are actions arriving from one level up in the agent's stack of predictive control modules. Adaptive goal distribution strategies, that is, modules controlling which goal is chosen and when, is an exploration and learning problem itself, just like sensorimotor learning at the proprioceptive level, at least in principle. Such strategies are commonly known as motivation.
Several research questions questions are attached to the teleology of artificial agents and robots. Exploration is the basis of all learning, and in some environments exploration alone is sufficient for agents to survive. On the other hand, there is no clear limit on complexity when considering all possible environments. An agent with limited resources, needs to handle unexplained phenomena with *active uncertainty*, which is randomness that is controlled in a precise fashion. There are many unanswered questions about the interplay of model, learning, and exploration, and having a theory of exploration strategies would be highly desirable. In some niche regions of the space of all environments, there exist problems, whose optimal solutions imply and require controlled uncertainty (Loeb 2012; Iigaya et al. 2017). Foraging (Charnov 1976) provides a very interesting example of open context problems of adaptation to non-stationary distributions. The challenge lies in the fact, that these problems cannot even in principle ever be fully explored. This is reflected by a huge number of different and often unexpected behaviours of foraging animals (Dugatkin 2014).
Another question is about the required introspection skills of local learners and corresponding introspective error statistics which can be used to modulate ongoing hierarchical cooperative learning. Introspection refers to the means a learner has to measure its own internal state. The total state is not only composed of raw sensory measurements of the external environment, but also by somatic sensors and integrating compound states computed from direct measurements, such as the available energy budget, the available adaptive capacity, integrated estimates like Bayes

filters, error statistics, and so on. It is clear that autonomous learners need to use introspection to modulate their overall learning.
All sensor, motor and internal state values are combined into one large state vector $\boldsymbol{s} \in \mathbb{R}^{\dim}$, with dimension $\dim = |\mathrm{A}|$ counting the number of variables defined by agent A. These are sampled synchronously at equidistant discrete times $t_{k\in\mathbb{N}}$, resulting in a new state $\boldsymbol{s}_t$. After storing the current state by appending it as a row to the $k-1 \times \dim$ matrix $\boldsymbol{S}$, the agent computes a set of functions of the state. This set consists of three elements at least, computing the current error $\boldsymbol{s}_t^{\mathrm{err}} = \boldsymbol{s}_t - \hat{\boldsymbol{s}}_t$, updating the internal state based on the error, and computing a new prediction $\hat{\boldsymbol{s}}_{t+1}$. Let $\hat{\boldsymbol{s}}_{t+1,\mathrm{prop}}$ denote the proprioceptive part of the prediction with the index set $\mathrm{prop} = \{j | \boldsymbol{s}_j \text{ represents proprioceptive channel}\}$. These channels are, by definition, wired to the motor units to control the motor primitive. This concludes one iteration of what is called the sensorimotor loop and the cycle is repeated.
Controlling motor primitives with proprioceptive predictions is a consequence of the predictive processing (Adams, Shipp, and Karl J Friston 2013; Clark 2015) model and is used here in replacement of conventional notations for action by letting

$$\begin{aligned} \hat{\boldsymbol{s}}_{t,\mathrm{prop}} &= \boldsymbol{m}_t && (3.1)\\ &= \boldsymbol{a}_t && (3.2)\\ &= \boldsymbol{u}_t && (3.3) \end{aligned}$$

This is an unusual notation but is done for practical reasons beyond formal coherence. It can significantly simplify the modelling structure by making motor and sensory pathways fully symmetric. It also provides principled cohesiveness of low-level models through shared representations among perception and action branches, that can be reused for different types of inferences in each model, for example in forward and inverse models.
The sensorimotor space $\boldsymbol{\mathcal{S}}$ is the vector space spanned by the vectors in the sensorimotor data matrix $\boldsymbol{S}$, written $\mathrm{span}(\boldsymbol{S})$. In general, this space is high-dimensional but the data in it is sparse and highly structured. This phenomenon known as the manifold hypothesis (Fefferman, Mitter, and Narayanan 2013; Mattingly et al. 2018; Li et al. 2018) and the full structure referred to as the *sensorimotor manifold*. Adaptive processes can be interpreted as discovery (exploration), and approximation (learning) of parts of the true sensorimotor manifold, which is usually unknown and represents an upper bound on the information that an agent can produce by interaction with the environment.
Sensorimotor models are treated as black box components within a developmental model (dm). The developmental model describes how the inputs and outputs of the component are connected to other variables. The black box is then configured with a particular learning algorithm for approximating the input - output relationship from the data. This allows to use any known supervised learning algorithm and to evaluate algorithms with respect to the component's required function by systematic variation. This represents a modular approach chosen for expected merits in representation (Nardi et al. 2006; Clune, Mouret, and Lipson 2013; Jr 2016). This is different from end-to-end methods used in many deep learning based approaches (Wahlström, Schön, and Deisenroth 2015; Punjani and Abbeel 2015) but certainly complementary as shown in (Hwang et al. 2017).

An important question in this context is about the model priors. In Bayesian terms the prior is well defined as a probability distribution, in robotics and other applied learning scenarios, the prior knowledge has to be translated into the hyperparameters of the model, including the model choice itself. Given finite data, the prior is a key parameter in learning generalization performance and robustness. This relation of *data* and *prior* in machine learning is shown in Figure 3.1, with possible design choices. A set of robotic priors has been proposed and evaluated with fitted Q-learning in (Jonschkowski and Brock 2015) using a reinforcement learning (rl) approach. Reinforcement learning is a computational theory of trial-and-error learning (Sutton and Andrew G. Barto 1998), that was inspired by conditioning models of animal learning, such as the Rescorla-Wagner rule (Rescorla and Wagner 1972).

Many value learning algorithms do not scale well to application in continuous state-action spaces and costly exploration, as is the case in robot learning. The most successful family in this domain with state of the art results is that of actor-critic algorithms. In these algorithms, the policy is a parameterized function approximator, is represented separately from the value function itself, and the policy space can be searched via the parameters for large returns. Directed search like policy gradient algorithms is often particularly efficient (J. Peters and S. Schaal 2006; Kober and Jan Peters 2011; Grondman et al. 2012).

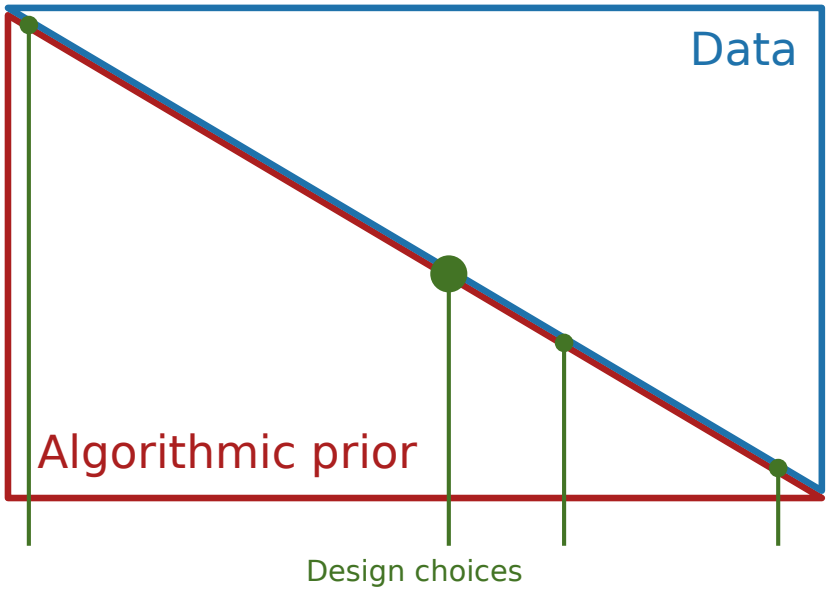



Figure 3.1.: Schematic relationship of algorithmic prior and data in machine learning and robotic learning problems. Assuming that the prior was selected appropriately, the stronger it is, the less data is needed for an equivalent amount of generalization. Diagram reproduced from memory of a presentation by Oliver Brock (2012).

A fundamental model of neuronal learning mechanisms is Hebbian learning (Hebb 1949), often abbreviated as "what fires together wires together", is written

$$\Delta w \quad = \quad \eta \cdot x \cdot y \tag{3.4}$$

This associative update rule is self-amplifying and thus unstable in its basic form. Besides local inhibition, several stabilizing factors have been proposed, for example in (T. Sejnowski, Chattarji, and Stanton 1989; Butko and Triesch 2007; Van Rossum, Shippi, and Barrett 2012). If the modulating factor is interpreted as a reward, it is shown to be compatible to temporal difference learning (Kolodziejski et al. 2008). This can be extended to more detailed models of synaptic modification such as spike timing dependent plasticity (Gerstner et al. 1996; Rao and T. J. Sejnowski 2001). The positive outlook on the biological plausibility of the current approach is concluded by considering the evidence in favor of the reward prediction hypothesis of dopamine (Wolfram Schultz, Dayan, and Montague 1997; Dayan and Niv 2008; Niv 2009).
In contrast to optimality, the principle of adequate design (Rashevsky 1973) suggests adequacy instead of optimality (Loeb 2012) as a general objective for driving adaptation. The return or loss is in many cases only an approximation of the true loss, so an optimum should be carefully weighted (variance problem). In open-ended learning it is actually necessary for the agent to be able to come up with new tasks and attached losses on its own, making losses and optima transient. Adequacy is about populations of solutions and encourages diversity, which is known to beneficial for ongoing adaptation in non-stationary environments, for example increased population robustness. In bootstrapping scenarios, the time needed for learning an adequate solution can clearly be a major criterion.
So far the term behaviour has been used without any definition. For completeness one is provided now.

> Given any object, relatively abstracted from its surroundings for study, the behavioristic approach consists in the examination of the output of the object and of the relations of this output to the input. By output is meant any change produced in the surroundings by the object. By input, conversely, is meant any event external to the object that modifies this object in any manner.

On this basis, "any modification of an object, detectable externally, may be denoted as behavior" (Rosenblueth, Wiener, and Bigelow 1943). Behaviour is used here in a very general way as anything observable about any discernible entity that it is coupled to its surroundings. This is followed by an alternative but compatible agent definition as an object with active behaviour, which

> is that in which the object is the source of the output energy involved in a given specific reaction. The object may store energy supplied by a remote or relatively immediate input, but the input does not energize the output directly.

Energy is a resource of fundamental concern for any real world agent. This includes all known organisms and robots. In particular, it includes robots that should be autonomous in the long-term. An actual time span in seconds is not given, since this depends on the agent's scale and corresponding time constants. Adaptation is shown to be a necessary consequence of resource

constraints (Otto E. Rössler 1974). The challenges posed by an environment E to an agent A in terms of survival are distributed over a wide range of diffculty. An example of a taxonomy is provided by the navigation hierarchy in (Franz and Mallot 2000). With the current definition of behaviour, this taxonomy can be extended to spatial problems beyond those of navigation. Motion skills are the ability to go from one location to another in the sensorimotor configuration space while respecting constraints on the path taken in between.

In many environments, the agent state can be adapted quickly and this can be done so repeatedly by innate controllers (Ashby 1952). In other cases, an agent's fitness in a given environment is improved by remembering locations and being able to quickly get there again. The mechanisms of formation, refinement, and persistence of memory are referred to as learning. Learning processes come in large variety, but they seem to be built into organisms from the lowest level. There is a lot of evidence in favor of learning processes being present not only in humans and animals, but also in plants (Trewavas 2014), single cells (Saigusa et al. 2008; Bray 2009), and possibly below (Monod 1972).

A large number of algorithms are known for learning the parameters of a model from data. The algorithm effects that the learned parameters will explain the data an in an optimal way with respect to some measure of fitness. The number of parameters is usually significantly lower than the number of data points. Supervised learning algorithms represent a particularly efficient class of learning algorithms, but these can only be applied if the data is presented in a suitable way. In machine learning, suitability is provided by a data scientist, but an autonomously learning agent needs to find a suitable represention on its own, choose an objective and decide, what data to use.

# 4. Structure of the thesis

The part following this one is the *main body* of the thesis and consists of three chapters. The Sensorimotor experiments chapter introduces the framework that was accumulated and used in preparation of this work. The experimental framework is based on recent sensorimotor theory and uses a graph based language for configuring sensorimotor experiments as workflows. These are expanded into an actual computation graph, that is then run autonomously. Almost all embodied learning agents discussed in this text will be represented, evaluated and analyzed using this approach. The most recent software version is the Python library *smp_graphs*, which is available on the internet (Berthold 2018b). The library contains the configuration of each experiment in the main part of the thesis.
The Self-exploration chapter motivates and introduces *self-exploration* as an explanatory tool for adaptive systems research. A working definition of *self* is given and placed in context with agent, body, and environment. Within this picture, exploration is discussed as an inside-out growth process. The rest of the chapter presents two major contributions of this thesis, which describe and quantify a probabilistic graphical approach to systematically describe how *function* emerges from *data* in adaptive sensorimotor models.
In the final chapter, a general developmental model for Skill acquisition in embodied agents is constructed from the preceding results. The model is illustrated on the basis of three different variations of that model, which are evaluated and compared on identical systems. Each of the variations is shaped by approaching the problem from the point of view of forward-inverse model pairs, predictive processing, or reward-based learning perspective. The thesis concludes with a summary and outlook.
**List of contributions**

- **tappings**, a graphical model and systematic approach for prediction learning on sensorimotor data,
- **infoscans** or quantitative tappings, a learning algorithm for tappings based on scanning information theoretic dependency measures,
- implementations of several online low-level learning algorithms. This includes versions of the exploratory Hebbian (EH) learning rule, recursive least squares (RLS), first-order reduced and controlled error (FORCE), Hebbian associative self-organizing maps (HebbSOM), and
- **extensions** of these algorithms for delayed output / input relations, eligibility traces, and an eligibility-based **delay estimation** technique,
- developmental **models for skill acquisition**, based on forward-inverse model pairs, predictive processing, and reward-modulated approaches,

- a **workflow language** for systematic design of sensorimotor learning experiments (smp_graphs) along with examples and documentation.

# Part II.

# Self-exploration and skill acquisition

# 5. Summary

This part provides the main body of the thesis in three chapters and contains a presentation of the major lines of work, and some resulting contributions to the state of the art in the fields of developmental robotics, and embodied artificial intelligence. The presentation consists of a formalization of sensorimotor experiment workflows and their mapping onto computation graphs with a focus on function reusability and self modification in Chapter 6, a provisional definition of an agent self and incremental results on the self-exploration hypothesis, the exploration of its own self-region by the agent, as a complementary process in the simpler picture of monolithic "learning" in Chapter 7, and a set of functional developmental models from three different conceptual families that are capable of single-episode, incremental inside-out exploration, learning, and skill acquisition in Chapter 8.

# 6. A sensorimotor framework

> Experiment is the sole source of truth.
>
> Henri Poincare, 1905

In this chapter, relevant parts of current sensorimotor theory will be introduced. The framework of this thesis will be developed from existing concepts and extended where necessary, both by *incremental* modification of existing concepts, and by addition of novel ones. This is done in three major steps, with each step graphically illustrated along the way. The aim is to obtain a view of the agent - body - environment interaction based on the information flow (Max Lungarella and Olaf Sporns 2006) that occurs between them. The information perspective enables a complementary understanding of how function and inference can be decomposed within networks of adaptive models that represent agent brains. The framework is then translated into a graphical scientific workflow language that is used in a series of experiments for providing a minimal but computationally and algorithmically complete explanation of low-level sensorimotor learning. This will provide the basis for what is presented in the consecutive chapters on self-exploration, and skill acquisition.

## 6.1. Sensorimotor experiments

A common picture is an agent A interacting with an environment E via actions $\boldsymbol{a}$ and state $\boldsymbol{s}$, shown in Figure 6.1.1). In reinforcement learning literature, this often includes a reward $r$ as a separate mode of environmental feedback in addition to the state measurement. The reward is omitted here and included without loss of generality in the state $\boldsymbol{s}$.
The first modification is shown in Figure 6.1.2), and consists of wrapping the agent A inside a body B. This follows from the embodiment hypothesis, and the outcome is an *embodied agent*. As a result, the action $\boldsymbol{a}$ emitted by the agent is shown as being transformed into the body reference system as action $\boldsymbol{a}_{\mathsf{B}}$. The body's action is then transformed into corresponding environmental activity $\boldsymbol{a}_{\mathsf{E}}$, eventually leading to a environmental state $\boldsymbol{s}_{\mathsf{E}}$, which has to travel through the body, becoming $\boldsymbol{s}_{\mathsf{B}}$ and to reappear as the agent's state $\boldsymbol{s}$.
This is followed by placing the embodied agent *strictly* inside of the environment. The information flow is rearranged in parallel, highlighting that action and perception are very closely linked. In the predictive processing view, this is taken one step further by stating that action and measurement quantities are really about the same thing altogether, and reinterpreting the action $\boldsymbol{a}$ as a prediction $\hat{\boldsymbol{s}}$, and the measurement feedback as a prediction error $\boldsymbol{s}_{\mathsf{err}} = \boldsymbol{s} - \hat{\boldsymbol{s}}$. In the corresponding diagram, shown in the bottom left of Figure 6.1.3), the notation is kept consistent for clarity, but the predictive processing interpretation will be repeatedly referred to.

The final step is to focus on a radial cut from the agent at the center to the environment. The region around the cut line is shown as a window in Figure 6.1.4) and the new situation is shown in Figure 6.2. Agent, body, and environment appear as layers from left to right in the last diagram. This is proposed here as a novel way of visualizing agent - environment interaction, based on several recent proposals in the direction of an information based view on sensorimotor activity and learning. It provides the advantage, that phenomena of varying delays, temporal spread of action responses, and the lateral diffusion of action information into neighboring channels can be very well visualized, as is done here with two circular flows sharing place around the origin of the motor activity, and branching off each other to return with different lengths. The current picture is coarse but details can be filled in as needed.

As an example consider the scenario shown in Figure 6.3 using information packets traveling with the flow. The agent emits an action shown as packet 1. It is transformed into physical motion of the body, constrained by morphology and environment, measured immediately and fed back on the proprioceptive channel as packet 2. A copy of packet 2 causes additional effects through causal interaction with the surroundings, which are mixed with extrinsic noise and producing packet 3. Observing this with another different delay on the exteroceptive channel, for example vision, produces packet 4. Momentary motion will also cause additional changes in proprioception, but for clarity this is not shown in the figure. All packets arrive at different times, and the learning agent needs to figure out, which action packet they belong to, and how that action needs to be modified to make the returning packets have the shape of the original action 1.

On the left part of Figure 6.2 an intrinsic process is shown in light blue, which is assumed as some inherent drive to activity by the agent. The diagram highlights, that self-exploration is a topological consequence of embodiment. The agent needs to negotiate its way out through the body in terms of the amount of control, and thus predictability, it can acquire on the body.

In many environments, very simple uniform random exploration can be sufficient for the agent to survive. Examples are cellular movement, which is often random, or the homeostat (Ashby 1952), which relies on the fact that random reconfiguration can be performed at very fast rate. An in-depth exposition of exploration issues like coverage and reachabilty can be found in (Benureau 2015).

With increasing complexity of the environment it becomes exceedingly hard for simple exploration strategies to succeed. This means, that if a solution was found at large cost, the agent has a clear advantage from remembering the solution. This is open-loop exploration and learning, also called motor babbling (Demiris and Dearden 2005; Schillaci and V. V. Hafner 2011; Benureau, Fudal, and Oudeyer 2014) or off-policy learning. If exploration and learning are put into closed loop, a new class of adaptive behaviour arises from the ongoing interaction between exploration and learning. In the closed loop, the exploratory motor signal is taken as a sample from the output distribution of the controlling model. The most recent feedback is used to update the model and thereby change the exploration through the adapted output distribution, used for example in goal babbling (Matthias Rolf, Steil, and Gienger 2011; M. Rolf and M. Asada 2015) and on-policy algorithms like SARSA (Rummery and Niranjan 1994) or actor-critic algorithms (V. R. Konda and Tsitsiklis 2000) (Hasselt and Wiering 2007; Grondman et al. 2012).

The programme of robot skill learning pursued in this thesis, is integral to sensorimotor theory not only for understanding adaptive motor skills themselves, but also for sensorimotor accounts

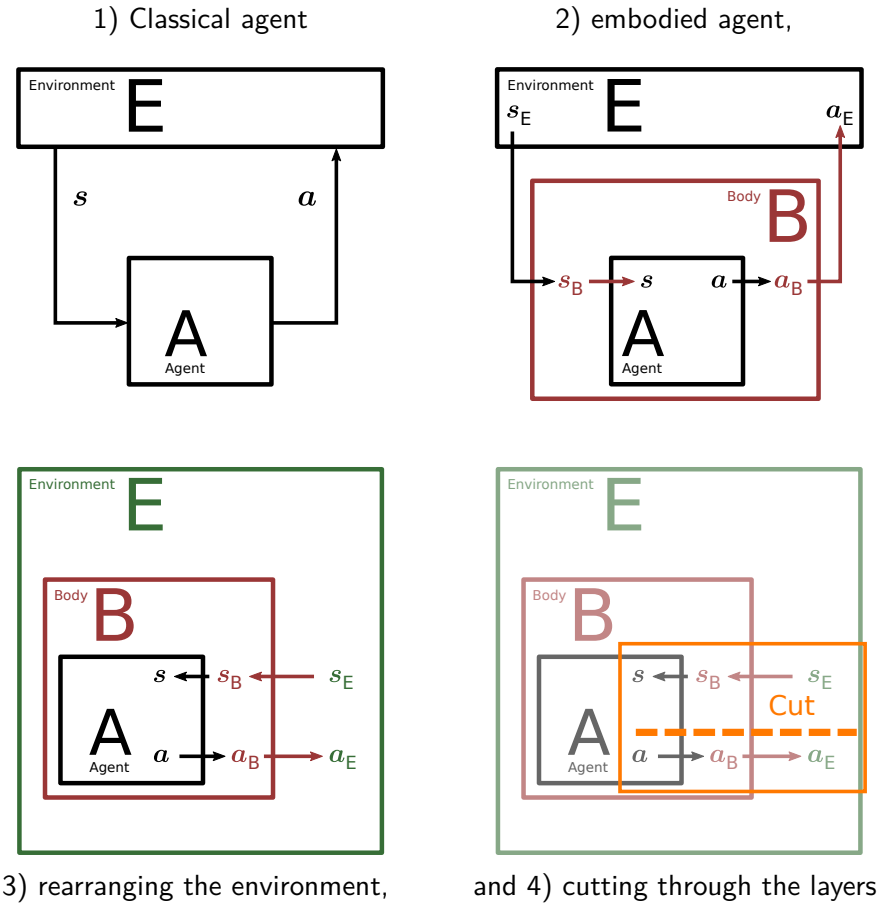



Figure 6.1.: Constructing the starts at the top left in panel 1) with the classical agent - environment diagram, making the agent embodied in the top right panel 2), properly placing the embodied agent inside the environment in the bottom left panel 3) and then isolating a region along an imagined line from the agent to the environment for examination. Obviously, the graphical sequence shown here is topologically trivial and is only provided as a supplementary line of arguments in favor of an information based picture of the sensorimotor interface.

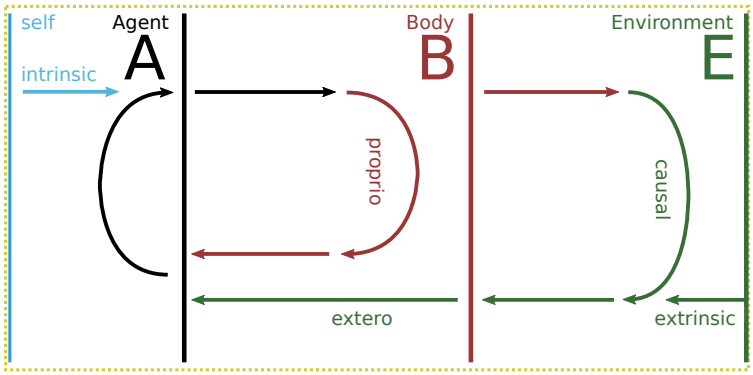



Figure 6.2.: Close up view of the cut showing the agent on the left in black, the body in red and the environment in green and the information flow occuring among these three layers.

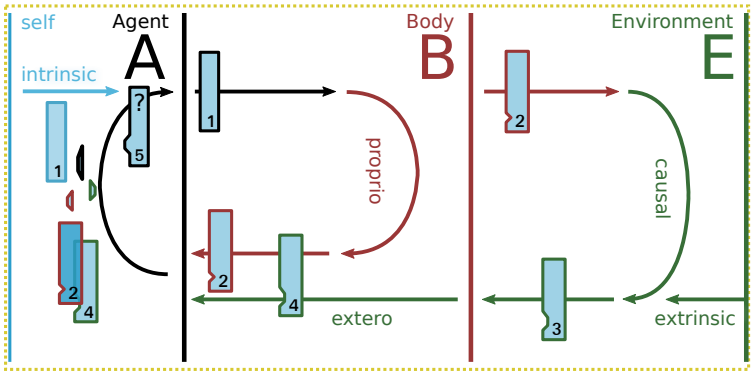



Figure 6.3.: Illustration of the body's and the environment's effects on information packets which the agent emits via its actions. Each layer adds its own characteristic changes to the information travelling through. The agent finally receives several modified copies of the original information packet. The agent uses the combined information available from all corresponding packets taken together to adapt its actions.

of cognition. Any such account must provide an explanation of the grounding in motor space. Research on cognitive aspects of mathematical thinking strongly suggests that also high-level cognitive function is built from the same circuitry as motion control and navigation functions (Lakoff and Nunez 2000), proposed less rigourosly earlier in (Lorenz 1973). Innnate reflexes and corresponding reflex arcs constitute the immutable building blocks of all behaviour. In the dynamic systems approach, motor primitives are thought of as adaptive versions of reflexes (Hogan and Sternad 2012) and a lot of research has focussed on the use in robotics (A. J. Ijspeert, Nakanishi, and Stefan Schaal 2002; Stefan Schaal, Jan Peters, et al. 2005; Konczak 2005; A. J. Ijspeert 2008), establishing a connection to control theoretic approaches (Stefan Schaal, Mohajerian, and A. Ijspeert 2007; Lewis 2009), and contributing to state of the art results in robot motion learning (J. Peters and S. Schaal 2006). The integration of sensory feedback into the primitive is only considered in some of this work (Kober, Mohler, and Jan Peters 2008). Nonetheless, a full sensorimotor version of primitives and unsupervised learning techniques have been proposed early on (Todorov and Ghahramani 2003; Mussa-Ivaldi and Solla 2004). This approach has also been pursued as part this work (Berthold 2018b; Berthold 2018a; Berthold and V. Hafner 2017; Gerken, Berthold, and V. Hafner 2017; Berthold and V. V. Hafner 2015; Berthold 2015; Berthold and V. V. Hafner 2014; Berthold and V. V. Hafner 2013b; Berthold and V. V. Hafner 2013a; Berthold, M. Müller, and V. V. Hafner 2011; Berthold 2011; V. V. Hafner et al. 2010; Berthold 2009; Horn 1986; Schunck and Horn 1981; Berthold and V. V. Hafner 2013c; Berthold and V. V. Hafner 2013a; Berthold and V. V. Hafner 2014). This can be seen as very particular instances of sensorimotor contingencies (O'Regan and Noë 2001; Buhrmann, Di Paolo, and Barandiaran 2013), a central notion in the sensorimotor approach. On a coarse level sensorimotor contingencies encapsulate the idea of internal models and prediction learning by defining the contingency as any kind of regularity in the sensory response to action, that can in principle be learned through the observation of an agent's own activity and the corresponding measurements. An early approach to self-organizing behaviour, that was eventually merged to some extent into the sensorimotor contingency framework is proposed in Verschure, Kröse, and Pfeifer 1992. A predictive processing formulation of sensorimotor contingencies is provided in Seth 2014.

Since the inception of information theory (Shannon 1948; Wiener 1949) it was developed to large extent within physics, but has more recently gained a lot of attention in the study of adaptive behaviour and learning, resulting in several novel proposals that firmly establish information based approaches in the field, and leverage it to move beyond traditional stimulus-response driven views of behaviour (Clark 2015), or overly simple optimization based explanations of exploration and learning (Polani, Olaf Sporns, and Max Lungarella 2007; Karl J. Friston and Stephan 2007; Klyubin, Polani, and Nehaniv 2008; Lehman and Stanley 2011; Salge, Glackin, and Polani 2013; Martius, Der, and Ay 2013).

Current quantitative information theoretic methods are an achievement of an ongoing line of research into sensorimotor interaction based on information flow. Starting with the explicit consideration of a model's embedding in sensorimotor context (Scheier, Pfeifer, and Kunyioshi 1998), a line of investigation emerged leading to novel insights for sensorimotor network analysis using quantitative methods in general (Max Lungarella, Pegors, et al. 2005), mapping information flows (Max Lungarella and Olaf Sporns 2006; Kaplan and V. V. Hafner 2006), and quantifying embodiment (Pfeifer, Max Lungarella, Olaf Sporns, et al. 2007; Polani, Olaf Sporns, and Max Lungarella

2007). This has also provided new ideas about self-organizing behaviour (Pfeifer, Max Lungarella, and Iida 2007), proposed earlier in (Der, Steinmetz, and Pasemann 1999), and more recent results in (Gershenson and Fernandez 2012; Martius, Der, and Ay 2013; Martius, Jahn, et al. 2014; Martius and Olbrich 2015), including work in direct relation to this thesis (Gerken, Berthold, and V. Hafner 2017).

In this framework, learning is done by fitting a model to data. When such models reside *inside* an agent they are referred to as *internal models* (Craik 1943). Internal models are a powerful concept used across the fields of control theory, neuroscience, and psychology (Tin and Poon 2005), and are thought to enable qualities in behaviour which are not realizable in a purely *reactive* agent. Internal models are not necessarily adaptive and can just as well be conferred to the agent a priori in the design process, which is done in most control theoretic approaches. Conversely, the current approach considers algorithms which are able to produce behaviour starting with blank adaptive models. The hypothesis is that any algorithm that can solve *this* bootstrapping problem can also solve relaxed versions. In other words, adaptation of behaviour at later points in the agent's lifetime can be *reduced* to adaptation from a null behaviour.

The agent's questions at this level are 'what do I need most urgently', 'to what extent do I know where to get it', 'to what extent do I know how to get there?', and 'what to do at different levels of uncertainty I might find?'. In order to provide algorithmic answers the problem needs to be put into detailed formal shape.

A minimal agent A consists of an internal structure $p$, encoded in the agent configuration $i$, predicting a sensorimotor state $\boldsymbol{s} \in$ the sensorimotor space span$(\boldsymbol{S})$, which is the combined state and action space. Actions are equally referred to motor commands, motor signals, or proprioceptive predictions. The mechanisms for interpreting these are innate reflexes and motor primitives. An important consequence of unifying actions with proprioceptive predictions is, that a pair of closely related prediction - measurement variables can safely be assumed, which not only facilitates exploration and learning, but constitutes the embodied bottleneck of an agent's grounding.

The component $p$ and its internal structure is the main subject of interest. Solutions to a specific robot learning problem will be different realizations of $p$, called *developmental models* (DM). Most importantly, DMs comprise a different level of modelling than sensorimotor models and are composed of *at least one* internal adaptive sensorimotor model. The DM specifies, which of the relations implicitly present within the sensorimotor data are learned and how the predictions (model outputs) are connected to other model inputs like the motor system. More interesting DMs consist of more than one senorimotor predictor. In the simplest arrangement, these predictors are linearly stacked on top of each other output-to-input. The lowest level $l_0$ is at the raw sensorimotor interface and each predictor further in represents its own level $l_d$ with $d =$ layer depth. Now the agent's *motivation* is encoded in $p_1$ whose predictions are goals which are fed as inputs to $p_0$. This means, goals should be sampled from a space close to input space of $p_0$. The raw prediction error resulting at $l_0$ is propagated back upwards through the stack to drive prediction and adaptation activity.

A configuration with two such components can already result in interesting dynamics, depending on their interaction. The prediction error can be reduced by changing any of the error operator's arguments, the prediction and the measurement. The prediction could be moved towards the

measurement independent of any change in motor output, and at the same time, the motor output could be changed to move the measurements it produces closer to the prediction. These two processes can interact cooperatively or competitively leading to different kinds of behaviour during and after learning. Observing prediction error statistics over time, an adaptive model can modulate its own operation by changing learning rates and other high-level parameters. The main error states a model needs to distinguish are 'learning and making progress' ($dE/dW \approx$ const), 'not learning and doing fine', 'not learning and getting worse, and 'learning but no progress'. Reward type sensorimotor variables can be encoded directly or derived from prediction error statistics. This makes the framework fully compatible with existing reinforcement learning methods.
From a fitness or survival perspective, the agent's task is to visit vital resources (goals), which are scattered in the environment according to some distribution. Visits need to occur frequently enough so that the agent never runs out of resources. This process is called spatial adaptation and implies motion. On the internal level this is fully captured by goals. Later on, it is argued that goals are samples from a predictive distribution, referred to as motivation. The unifying property of the survival and goal scenarios is indisputable necessity beyond any local control. In the fitness scenario, the goal recognition is computed by the body and environment alone, without the brain. In both cases the *task* can be posed for example as

$$P_i = \sum_t c_t^- + c_t^+ \geq 0, \qquad \forall i \in \mathbb{R}^+. \tag{6.1}$$

with $c^-$ being the momentary consumption, and $c^+$ the momentary acquisition of resources. The problem can be put in terms the gradient search which minimizes the expected $\bar{P}$ over an episode $\boldsymbol{S}$ of length $k$ by

$$\nabla_i \sum_t P(\boldsymbol{s}_t) \tag{6.2}$$

with all episodes recorded in $\boldsymbol{T}$ and considering all $\boldsymbol{S}_i$ satisying

$$\sum_t P(\boldsymbol{s}_t) \leq P_{\text{crit}} \tag{6.3}$$

which is the set of all adequate agents with accumulated cost less than zero over the episode. Solutions can be imagined as points in the cost-reward plane, with a purely illustrative example given in Figure 6.4. All solutions in the region bounded by the minimal cost to the west, the maximal reward to the north, and the cost-reward balance line to the south east. Adequate solutions form a narrow vertical band along the left border of the "nice" region, while optima are expected on the top right corner. This does not mean that the current approach is anti-optimal, but that the emphasis is placed upon the optimization *process* rather than a single final result.
In this chapter the conceptual framework in which experiments are designed, run, and analyzed has been motivated and defined. The thesis' main topics, self-exploration and skill acquisition, have been introduced in more detail. An agent based representation has been developed based on 'predictive coding' which allows to design, implement and analyze a large set of developmental models. This provides the preliminaries for the next chapters, in which experiments probing the framework's hypotheses will be presented and discussed.

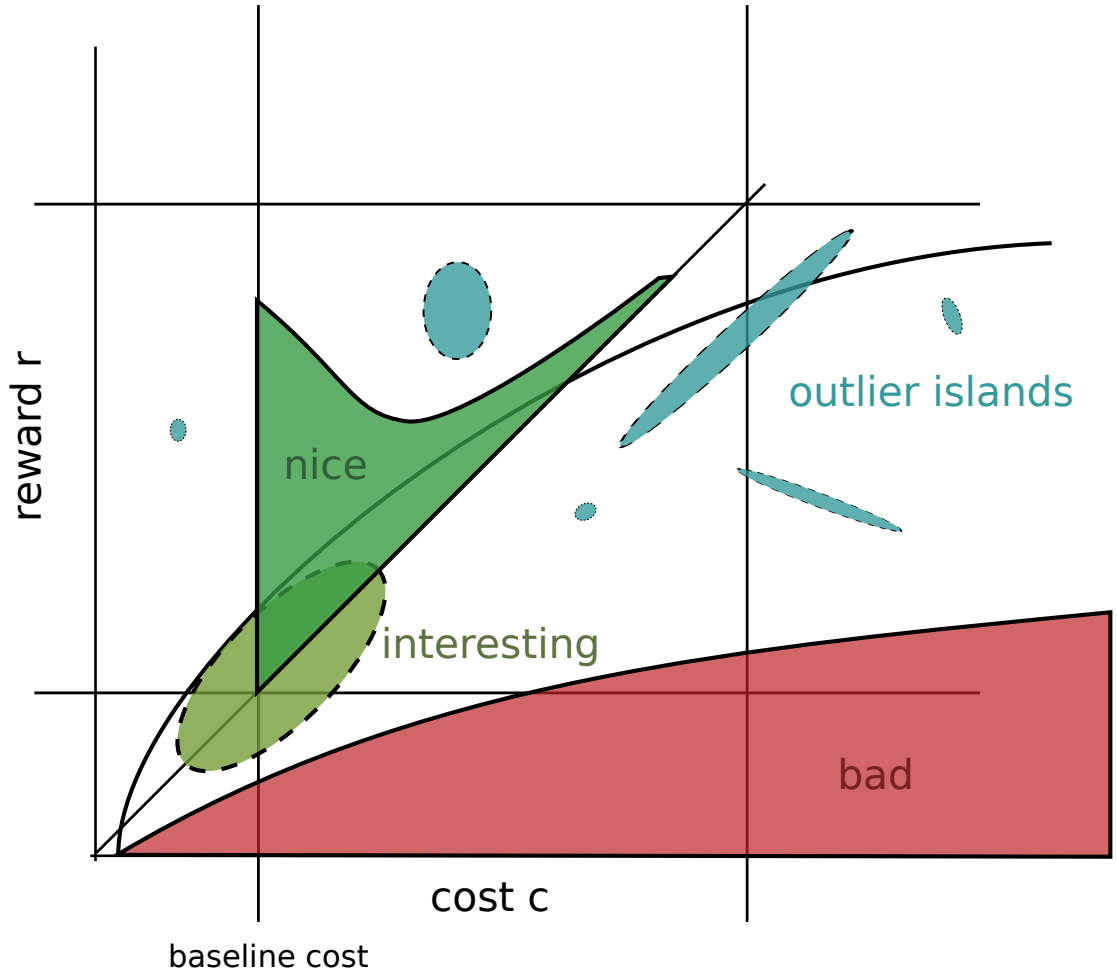



Figure 6.4.: Purely illustrative example of solutions distributed in the cost-reward space, indicating several different qualitative regions.

## 6.2. Software

These considerations have been captured in a formal way in the software project *smp_graphs*, available from (Berthold 2018b). This package is a complementary part of the thesis, and is documented thoroughly in the Software appendix, which is a verbatim copy of the software API documentation. For quick reference, the introduction section of the project's README is quoted below.

> **smp_graphs**
>
> This is an experimental framework for specifying sensorimotor learning experiments as a computation graph. The nodes represent functions which consume and produce signals which are the function's inputs and outputs and are represented as edges in the graph. This approach reflects the necessity of identifying design patterns in such experiments and capturing them in such a way that they can be reused across many different experiments. This idea is not new and smp_graphs simply represents the commitment to my own characteristic decompositions of the problems into reusable elements and patterns of arrangement. Of course there is no fundamental intrinsic restriction to sensorimotor learning so the framework can be used for any kind of computation flow. There are many examples of similar environments out there some of which I have used extensively and which acted as inspiration to my own design here. These are for example mdp, pylearn2, blocks, procgraph, keras, supercollider, puredata, gstreamer, gnuradio, and simulink / labview.
>
> The framework exists inside the larger sensorimotor primitives (smp) ecosystem and it implements only (mostly) framework specific functions of graph handling, manipulation, and execution. The actual algorithms are kept separately in a library called smp_base. Specific block implementations make use of other *smp_** libs, such as smp_sys (robots $\in$ systems) and other 3rd party python libs, see dependencies.
>
> An experiment's graph is specified in a configuration file written down as a Python dictionary. The configuration is then loaded by the general experimental shell 'experiment.py'. The assignment of values to a node's inputs is part of the graph configuration and is either a constant computed at configuration time or another block's output provided on a globally shared bus structure. Every block writes its outputs to that bus, where it can be picked up and used by any other block, including the block itself, allowing recurrent connections.
>
> The graph-based representation provides good separation of the experiment's algorithm and the implementation. The project is work-in-progress. In principle the configuration is independent of this specific implementation and could be run on other virtual machines than python. The most important drawback right now is the verbosity of the configuration dictionary. This can be improved and is planned to be done for future versions.

## 6.3. Random strategies

This section establishes baselines, against which models are evaluated with respect to how they perform in different environments. The choice of random strategies as a baseline is motivated using Bayesian priors. This identifies the baseline with the untrained state of adaptive models. The necessary measures are introduced for quantifying the effectiveness of sampling and adaption. A set of experiments reproduces established results to illustrate the behaviours produced by random strategies, and to highlight their limits.
The intention of this and the following sections and the sequence of experiments presented in there is three-fold. For one, it is meant entirely as a graphically illustrated account of the causes and consequences of divergence in sensorimotor systems for readers not familiar with these facts. The experiments do not represent any kind of new results whatsoever, but only reproduce known findings. For those indeed familiar with divergence and its compensation through adaptive models, it should help to arrive at a rough intuition about how divergence is reflected in probability based measures.

### 6.3.1. Baseline behaviour

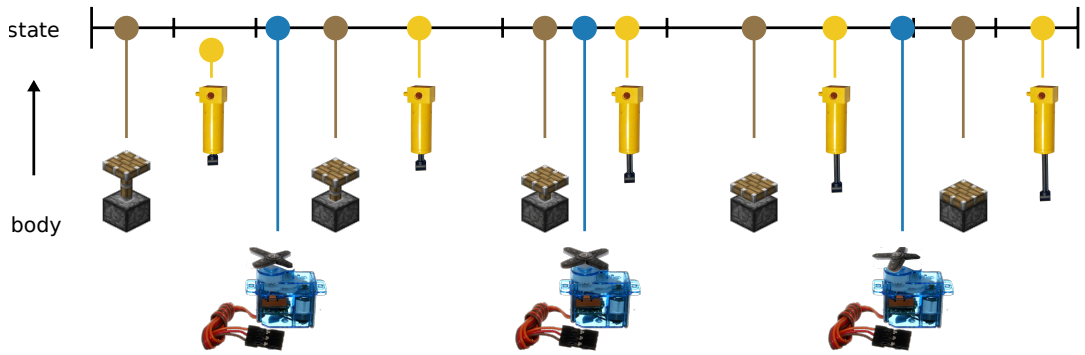



Figure 6.5.: The line labelled state at the top is the one-dimensional state space of the agent. The mapping of internal states to physical configuration is shown below for each of three different motor units.

The main subject of these experiments is an embodied learning agent, which is discussed in the introduction part and in more detail at the beginning of this part. In the text this is abbreviated interchangeably to agent, brain, or robot. The term system is used here equivalently to body and embodiment alone or both body and environment to distinguish from pure agent properties. Episodes of agent activity produce behaviours which are represented as multivariate timeseries of observations produced by continuous state measurements. Behaviours are analyzed and explained by considering the combined effects of information loops flowing through the brain-body-environment coupling.
Consider a simple body with one degree of freedom, basically a single motor. This dimension is mapped onto the agent's internal state space. A few examples of single motors in different configurations are shown in Figure 6.5 along with exemplary mappings of the motor state onto the state space. The ground truth state of a state measurement $s$ is

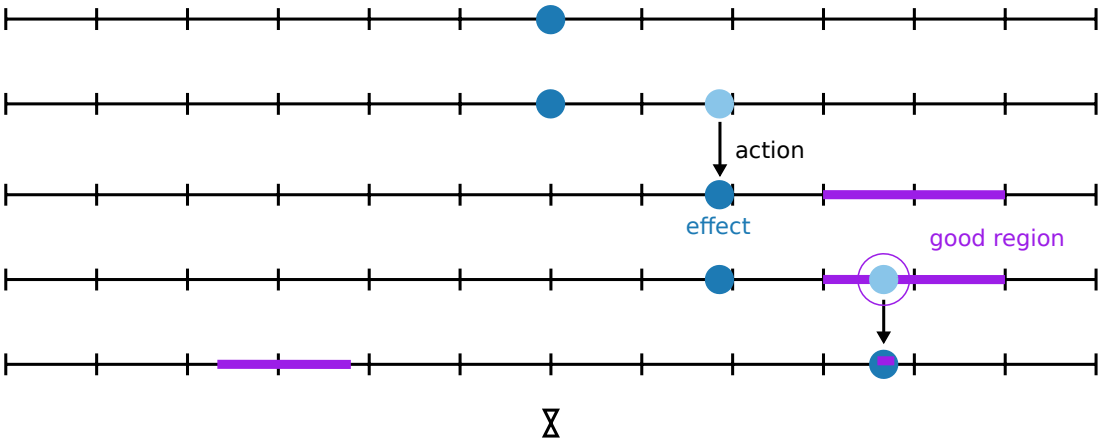



Figure 6.6.: Again, a one-dimensional state space is shown with time progressing from top to bottom. This example episode starts in a random initial state in the center of the first row, shown as dark blue circle. Next, an action is computed (row 2, light blue), which is executed resulting in a new state in the third row, where a favorable region of the space appears close to the agent's current state. By incidentally computing an action in row four that hits the goal, the resource is consumed and another appears.

$$p(\boldsymbol{s}) \sim p(\boldsymbol{s\prime}) \tag{6.4}$$

with $\boldsymbol{s\prime}$ the ground truth state and $\boldsymbol{s}$ the internal state, for example a single sensor reading. The internal state representation per se is completely independent of the geometry and material of the motor. These parameters are only relevant as far as they show up in sensorimotor statistics. The effective state space of an agent tends to grow very easily. In this framework, the minimum is two dimensions, resulting from one motor channel and one independent sensory measurement of that motor channel, against which the intended results (the action) can be compared. The effective state space, which is the sensorimotor space, can be written as

$$\text{span}(\boldsymbol{S}) \subset \mathbb{R}^{k \times \dim}$$

The state $\boldsymbol{s}$ is usually structured, for example through sorting the elements by their primary function, certain parts of the vector, or subspaces respectively, are indicated by a superscript, and time is indexed in the usual subscript way, for example the proprioceptive state at time $t$ is written $\boldsymbol{s}_t^p$. Both $t$ and $p$ are vectors in the general case, referring to selection masks along the time axis and the modality axis of a sensorimotor data matrix. The complete state at any time $t$ is expressed as

$$\boldsymbol{s}_{t+1} = \mathsf{A}_i(\boldsymbol{s}_t) \oplus \mathsf{B}(\boldsymbol{s}_t) \oplus \mathsf{E}(\boldsymbol{s}_t) \tag{6.5}$$

with the functions $(\mathsf{A}, \mathsf{B}, \mathsf{E})$ referring to lumped models of internal memory, body, and environment respectively using the names introduced in Figure 6.2. That is, each of these layers contributes something to the new state, in terms of information, based on a function of the most recent state, and all these contributions are combined by the $\oplus$ operator.

The motor commands are coded into the state vector as so-called proprioceptive channels. Proprioception means self-perception and is usually referred to as a sensation of the body configuration. By convention, proprioceptive channels $\boldsymbol{s}_p$ are hardwired to the motor units. A channel is a composite entity and expands into a fixed set of subchannels, each with a particular function, such as prediction, measurement, or error, each with respect to the same sensorimotor variable. These three functional types are indicated by $\hat{\cdot}$, $\check{\cdot}$, $e(\cdot)$, respectively. By associating the channel prediction with motor commands, and channel measurement with sensor readings, it can be assumed that the pair of variables is related by a mapping that is close to identity or at least monotonic over a wide range of values. This may seem innocuous, but it is a fundamental *axiom* of the agent's grounding in external reality. It provides a fixed point for pivoting synergistic relations among variables during all subsequent learning stages.
Different notations for motor signals, referred to alternatingly as motor commands $\boldsymbol{m}$, actions $\boldsymbol{a}$, or control inputs $\boldsymbol{u}$, are set equivalent to *proprioceptive state predictions* simply by writing

$$\boldsymbol{m} = \boldsymbol{a} = \boldsymbol{u} = \hat{\boldsymbol{s}}_t^p \tag{6.6}$$

Analogously, different names such as inverse model, controller, policy, and similar concepts are called *strategies* within the scope of the thesis. A strategy $A$ produces actions $\boldsymbol{a}$ by sampling a predictive distribution conditioned on the current state. More precisely, actions are proprioceptive predictions $\hat{\boldsymbol{s}}^p$, and the strategy is a predictive distribution

$$\hat{\boldsymbol{s}}^p \sim \mathrm{A}(\boldsymbol{s}) \tag{6.7}$$

with $\boldsymbol{p}_-$ and $\boldsymbol{p}_+$ the limits of proprioceptive space.
The current agent consists of a single proprioceptive channel, making it a kinematic system, where the variable measured by the sensor is completely causally determined by the most recent motor action, and nothing else. This is equivalent to the Markov property. This also means, that every reachable point can be reached within a single time step, without the need for prolonged travel towards a goal state. The body imposes an upper and lower limit of possible motion by geometrical constraints that should be evident. The environment in this example is simply free space and has no effect on the behaviour. By substituting into Equation 6.5 the system equation can be written as

$$\boldsymbol{s}_{t+1} = f_B(\hat{\boldsymbol{s}}_t) + \nu_0 \tag{6.8}$$

A very simple strategy is sufficient for solving the agent's task, which consists in sampling *uniform random actions* over the interval corresponding to its motor limits, which are assumed as prior knowledge, and without making use of feedback information. The task consists in finding and consuming enough resources to stay alive. The uniform strategy of the agent A can be written as

$$\mathrm{A}(\boldsymbol{s}) = \boldsymbol{\mathcal{N}}(\boldsymbol{p}_-, \boldsymbol{p}_+) \tag{6.9}$$

with $\boldsymbol{p}_-, \boldsymbol{p}_+ = \boldsymbol{s}_i, \boldsymbol{s}_j$ explicitly representing the lower and upper limits of the motor range by renaming particular elements $i, j$ of the complete state vector.
The probability of success for an agent with a fixed budget $B$ and goal density $\rho$ can be written as

$$p(\text{success}) = pr(k; n, p) \tag{6.10}$$

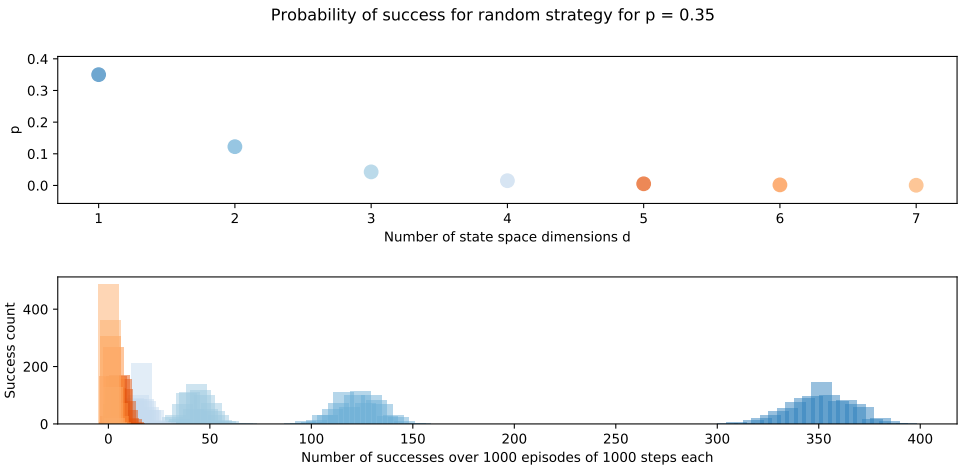



Figure 6.7.: Survival probabilities of random strategies for a kinematic system in state spaces of increasingly higher dimension and fixed goal size. The top plot shows the probabilities for 1000 trials over the number of dimensions on the x-axis. The top plot defines a color coding for each dimensionality from dark blue to orange. In the bottom plot seven histograms are shown using the same color coding. In the histogram, average budget values are counted over 1000 episodes of 100 steps each for each configuration.

which is a binomial distribution with parameters $k = 1, n = B$. The probability of a single success is equal to the relative density of goals in the space, so $p = \rho$ and the probability of a single failure is $1 - p$. Assuming a unit budget consumption rate, the probability of success is thus the converse probability of seeing more than $B$ failures in a single run of trials. To illustrate how quickly the probability vanishes, when the number of dimensions is increased, $p(\text{success})$ and the histogram of budget states is plotted in Figure 6.7 for a relatively large goal size of 0.1 units of space. This phenomenon is known as the *curse of dimensionality* and results from the volume's growth being exponential in the dimension and thus quickly outpacing the growth of *target* volumes. Empirical results from uniform random agent episodes are shown again in Figure 6.8, Figure 6.9, and Figure 6.10 in the next three experiments.

The situation only gets worse when the order of the system is increased, resulting in a *random walk*. An upper bound for the chances of a random walk hitting a single place on a lattice grid is given by the results on the return-to-origin problem on lattice grid random walks (Pólya 1921; Montroll 1956). In one- and two-dimensional grids the return is certain and rapidly decays in probability for higher-dimensional spaces (Weisstein 2018). The next few experiments examine the uniform random strategy on a kinematic system more closely.

## Experiment 1: Random agent

A very short episode (30 steps) of behavior of the baseline agent is demonstrated. The agent consists of the baseline strategy, performing open-loop uniform random search in finite isotropic space. The minimal function required for this behavior is *goal recognition*. This is modeled as regions around *target* points by thresholding the distance between state and target. The top plot in Figure 6.8 shows the goal position $\hat{s}_p^{l_1}$ as a thick blue line, the action $\hat{s}_p^{l_0}$ in dark green, and the resulting measurement $s_p$ in light green. The measurement is delayed by two time steps with respect to the action, highlighted by yellow causality lines for three action-measurement pairs, starting at time $t = 19$. The big red circles indicate points where the goal was met closely enough. The resource is consumed and another one appears in a random location. The bottom plots shows the time series of the agent's resource budget in units of the internal minimum resource consumption.

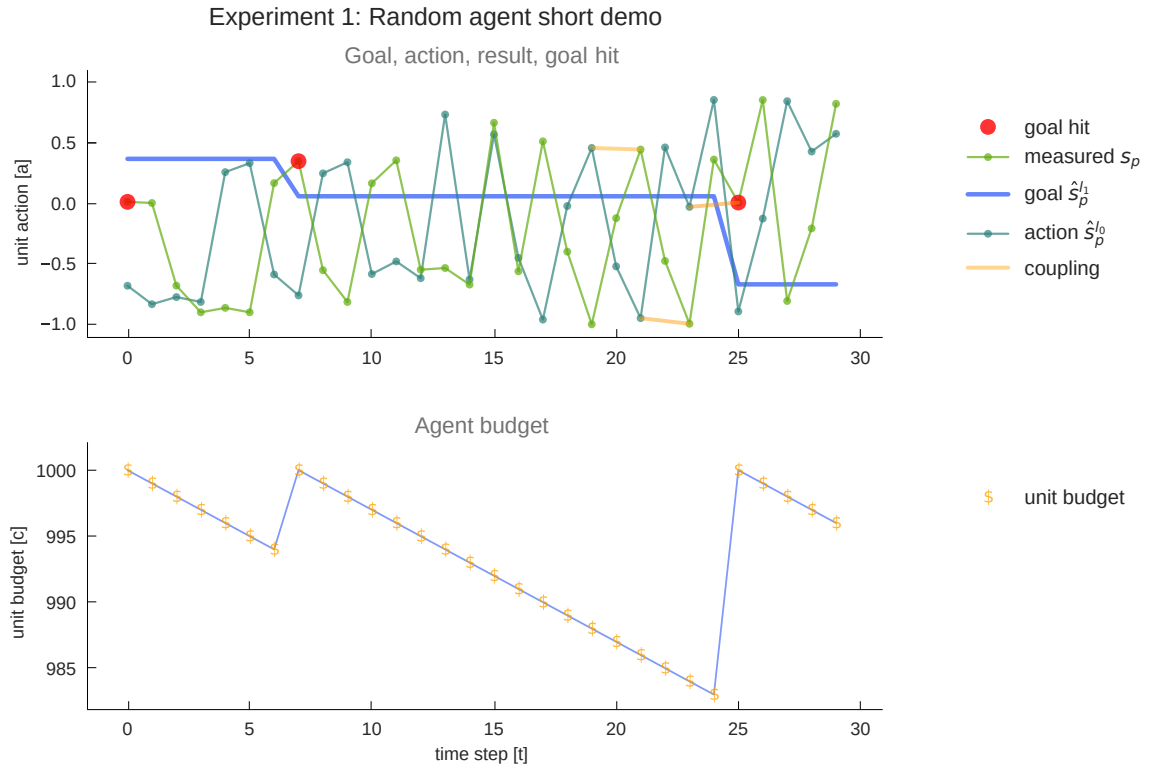



Figure 6.8.: Experiment 1-1 Illustration of the baseline agent behaviors. The top plot shows the goal position $\hat{s}_p^{l_1}$ as a thick blue line, the action $\hat{s}_p^{l_0}$ in dark green, and the resulting measurement $s_p$ in light green. The measurement is delayed by two time steps with respect to the action, highlighted by yellow causality lines for three action-measurement pairs, starting at time $t = 19$. The big red circles indicate points where the goal was met closely enough. The bottom plots shows the time series of the agent's resource budget in units of the internal minimum resource consumption.

The system used in Experiment 1 is a point mass system of zeroth order, denoted $\mathrm{pm}_0$. Zeroth order is the same as kinematic and means, that a state prediction (action) is directly transformed into a state measurement (sensation), applying only the inherent lag, and small amplitude motor noise in every time step, and also small amplitude noise on the system state transition matrix $\boldsymbol{W}_0$ once, at initialization. The full state update equation can now be written as

$$
\begin{aligned}
s_{t+1} &= \mathsf{E}(\mathsf{B}(\mathsf{A}(\boldsymbol{s}_t))) && (6.11)\\
\mathsf{B}(\boldsymbol{s}_t) &= h_0(\boldsymbol{W}_0 \cdot \hat{\boldsymbol{s}}_{t-lag}) + \nu_0 && (6.12)\\
\mathsf{E}(\boldsymbol{s}_t) &= \boldsymbol{I}\boldsymbol{s}_t && (6.13)
\end{aligned}
$$

with A as in Equation 6.9, and identity matrix $\boldsymbol{I}$. The system motor-sensor delay parameter is set to lag $= 1$. A lag of $l \in \mathbb{N}$ means, by convention of smp_graphs, that the measurement will appear $l + 1$ time steps later in the sensorimotor data stream, with respect to where the corresponding action appears.

### Experiment 2: Random agent full episode

This is a longer run of the same configuration as in Experiment 1, resulting in an episode of 2000 time steps of the baseline behaviour. The length of the episode is greater than the agent's initial budget. The strategy must statistically be good enough to let the agent survive for a number of steps larger than the initial budget, which would be consumed after the same number of steps when following a null strategy, that is, do nothing by a zero or constant action. The plot in Figure 6.9 is similar to the previous experiment. The histograms added on the right hand side show uniform and overlapping distributions for goals and goal hits in the top row and a large amount of mass close to the maximum for the budget values on the bottom row.

### Experiment 3: Random agent episode statistics

In this experiment the budget statistics over 20 runs of Experiment 2 are computed to illustrate the viability of the uniform random strategy in the low dimensional configuration. A histogram of the budget mean and minimum values is shown in Figure 6.10. The mean is close to the maximum value of 1000 and the minimum values are well above 900 and thus not critical.

### Experiment 4: Random agent dimension statistics

This experiment is an extension of Experiment 3, computing the same budget statistics over 20 episodes each, this time for all configurations of the sensorimotor dimension $d = [1, 2, 3, 4, 5, 6, 7]$ The results is shown as an error bar plot in Figure 6.11, which is in qualitative agreement with the data shown in Figure 6.7.

Experiments 1 through to 4 serve to illustrate a known result, that the baseline agent's success depends on the density of goals, with respect to its motor space. Assuming a constant unit-sphere

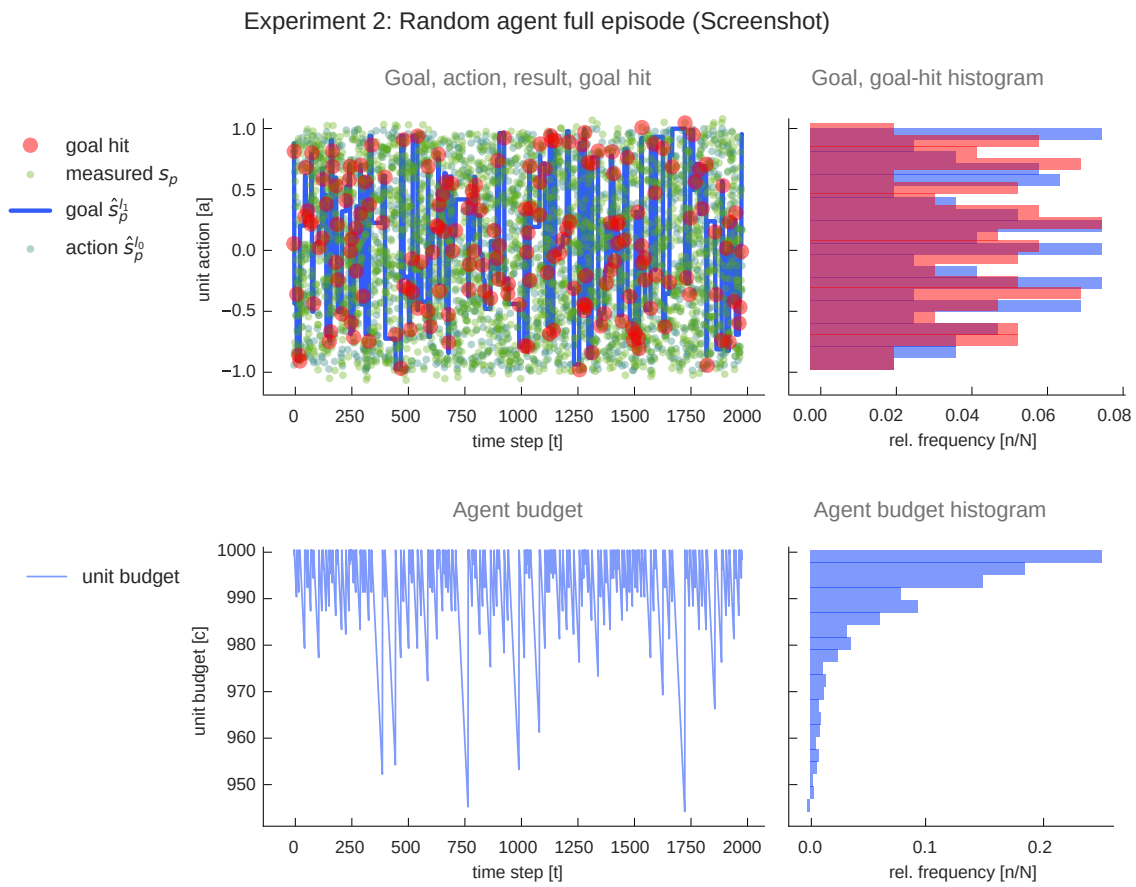



Figure 6.9.: Experiment 2-1 Screenshot of a full episode of the baseline agent behaviour covering an episode length of 2000 time steps. In the top left, the raw sensorimotor timeseries is shown, and in the top right the histograms of goal hits is plotted on top of the unique goal histogram, showing no obvious mismatch. The bottom row contains the same types of plots but for the budget variable, which never even gets close to a critical value.

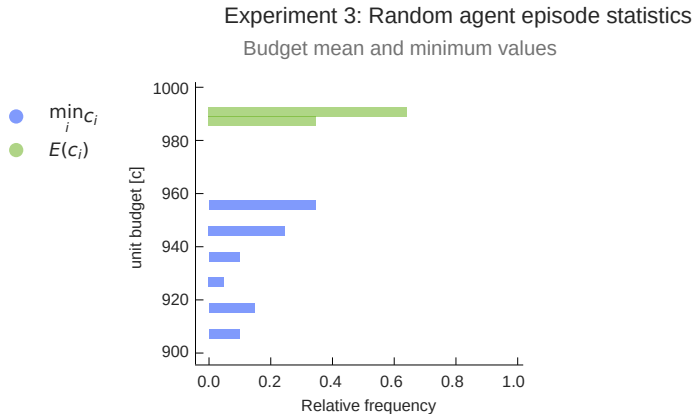



Figure 6.10.: Experiment 3-1 Statistics over 20 runs of Experiment 2, showing the mean, and minimum budget values during each episode in a combined histogram. The mean close to the maximum of 1000 and even the minimum values are above an uncritical value of 900.

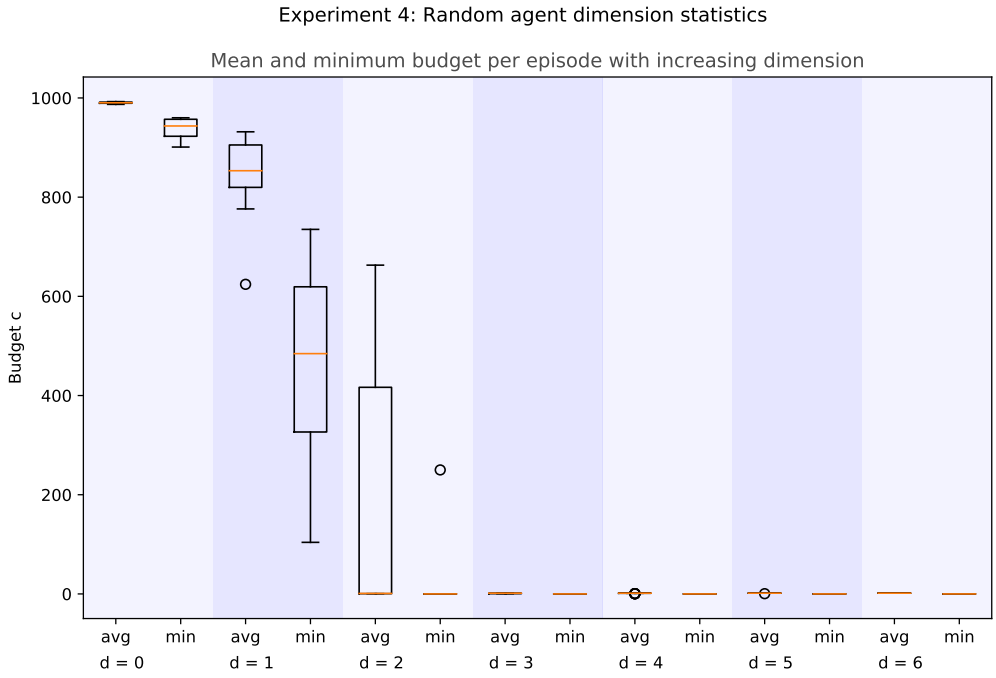



Figure 6.11.: Experiment 4-1 The budget statistics as in Experiment 3 for each configuration of the sensorimotor dimension $d$ with $d \in [1, ..., 7]$, increasing from top to bottom. This picture tells the same qualitative story as Figure 6.7, where the zero order random search fails with increasing dimension.

goal size and close to identity relationship of motor to sensor, the relative goal size decreases with dimension $d \in \mathbb{N}$, making the agent fail above some critical dimension $d_{\text{crit}}$. The precise value of crit depends on a set of additional parameters, which fully specify the details of agent, body, and environment (motor ranges, size and boundedness of space, budget magnitude and consumption rates, etc).

## 6.4. Divergence and information distance

The next few experiments elaborate further, how the probability of hitting goals that are uniformly distributed in the goal space also decreases when the agent's *access* to the goal space deviates from the identity mapping. The access is influenced by the amount of divergence between the motor and goal space distributions, which is greater than zero for all real systems due to the accumulated effect of small imperfections. Let $p(\hat{\boldsymbol{s}})$ denote the predictive distribution, and $p(\check{\boldsymbol{s}})$ the measurement distribution, of the same variable $\boldsymbol{s}$. Then this can be written as

$$\begin{aligned} p_{\text{success}} &= p_{\text{hit goal}} \cdot p(\hat{\boldsymbol{s}} \approx \check{\boldsymbol{s}}) && (6.14) \\ &= p_{\text{hit goal}} \cdot (1 - d(\hat{\boldsymbol{s}}, \check{\boldsymbol{s}})) && (6.15) \end{aligned}$$

so the agent would certainly like to make the right hand factor large, and conversely, keep the divergence $d(\cdot, \cdot)$ low, by a search process

$$\min_i d(\hat{\boldsymbol{s}}, \check{\boldsymbol{s}}) \tag{6.16}$$

over modified versions $A_i$ of itself. In the experiments, three terms or operators are used to model and control divergence by lumping together the causes of the agent, the body and the environment. The first one is instantaneous map distortion which performs a nonlinear transformation on the input through a parameterized transfer curve $h$, generating the measurement distribution via the state update rule introduced in Equation 6.12,

$$\begin{aligned} p(\hat{\boldsymbol{s}}) &= p(A(\boldsymbol{s})) && (6.17) \\ p_1(\check{\boldsymbol{s}}) &= p(h_0(\hat{\boldsymbol{s}})) && (6.18) \end{aligned}$$

The second one is memory and delay effects, which is present in any higher order system, and combined with transmission delays and coupling, this can easily lead to complex dynamic behaviour, resulting in a random walk or other type of $1/f$ statistics

$$p_2(\check{\boldsymbol{s}}) = p\left(h_0\left(\sum_m \boldsymbol{W} \cdot \boldsymbol{S}\right)\right) \tag{6.19}$$

where $\boldsymbol{W}$ is the $n \times m$ state update matrix, $\boldsymbol{S}$ is the $m \times n'$ sensorimotor data matrix, with $n' \geq n$ and $m$ the number of sensory modalities.

These two differ from the third factor of external entropy (EE). The environment is able to perturb the agent's state, while information is travelling from the agent's motors to its sensors. The true source can be a single non-deterministic external process, or the combined activity of many of them. In the first case, the perturbation clearly appears as nondeterministic noise to the agent. If the agent is using very simple models, deterministic behaviour will still appear as non-deterministic noise due to *under-modelling*. An example for this is hysteresis, a characteristic effect present in many real world systems. Hysteresis means, that a system will travel on two different trajectories, depending on the direction it is moving in. If this is not resolved by the agent, the overall effect appears in lumped form as unexplained entropy in

$$p(\check{\boldsymbol{s}}) \quad = \quad p\left(h_0\left(\sum_m \boldsymbol{W}\cdot\boldsymbol{S}\right)+\boldsymbol{\nu}_0\right) \tag{6.20}$$

with the noise term $\nu_0$ distributed according to $p_{\mathsf{EE}}$. Divergence is a statistical concept for comparing probability distributions, allowing to quantify distances in probability space. For the current scope, prediction and measurement are renamed to $X=\hat{\boldsymbol{S}}, Y=\check{\boldsymbol{S}}$. The Kullback-Leibler divergence (KLD, (Cover and Thomas 2006)) for example, is defined as

$$D(X||Y)=\sum p(x)\log\frac{p(x)}{p(y)} \tag{6.21}$$

for discrete probability distributions, and can be applied to histograms. Another frequently encountered divergence measure for histograms is the Chi square distance (Nielsen and Nock 2014),

$$\chi^2(X,Y)=\sum\frac{(x-y)^2}{x+y} \tag{6.22}$$

Another histogram divergence measure is the Earth mover's distance (EMD), informally the sum probability mass multipled by the distance it has to be moved, to make two distributions equal. Let $d(i,j)$ be the distance between bins $i,j$, and $f(i,j)$ the local flow between bins $i,j$. Then the EMD is defined (Rubner, Tomasi, and Guibas 2000) as the work

$$\mathsf{WORK}(X,Y,F)=\sum_i\sum_j d_{ij}f_{ij} \tag{6.23}$$

resulting from an optimal flow $F$ between the histogram bins $i,j$ of $X$ and $Y$, normalised by the total flow, giving

$$\mathsf{EMD}(X,Y)=\frac{\sum_i\sum_j d_{ij}f_{ij}}{\sum_i\sum_j f_{ij}} \tag{6.24}$$

This can be understood as an expected amount of adaptation activity necessary on the agent's side to match a given goal distribution, in rate-coded activity terms. A different kind of relative probability measures arises from considering the expected point-wise dependency of two variables. The most important representative is the *mutual information*, which can be defined as the Kullback-Leibler divergence of the variables' joint distribution and their product distribution (Cover and Thomas 2006),

$$I(X;Y) \quad = \quad D(P(X,Y)||P(X) \cdot P(Y)) \tag{6.25}$$

and which measures shared information in terms of statistical dependency. By complementarity, an *information distance* can be defined (Crutchfield 1990) as

$$d(X,Y) = H(X,Y) - I(X;Y) \tag{6.26}$$

which can be normalized to the interval $[0,1]$ by dividing through the joint entropy. These last two measures are based on information theory which will be considered in more detail in Chapter 7. These probability measures can be used as indicators for the underlying modelling effort required intrinsically by the data, regardless of the chosen model or learning algorithm.

### Experiment 5: Divergence measurement

This and the next two experiments illustrate the effect of the motor-to-sensor mapping on distance and divergence. This is done through controlling the /distortion parameters/ and the amount of /external entropy/, that is injected into the system. The effect is then measured using the root mean square error, the information distance, and the divergence between a sensorimotor state prediction (action) and a state measurement by a sensor. Each measure is taken as an average over the entire episode. Mixture parameters are introduced to the basic $pm_0$ system model which control the magnitude of these effects in the model system. The divergence is modelled by a transfer function defined over the interval $[-1,1]$. The response shape is controlled globally by sigmoid parameters and locally with colored noise. External Entropy is injected as point-wise independent noise on the sensor measurement, modeled as an additive noise term $u_t$. The principal senorimotor delay is controlled by the lag parameter. The experiment extends Experiment 2 by applying these measurements to the otherwise unmodified $\text{pm}_0$ system of Equation 6.12. The results are shown together with those of Experiment 6 in Figure 6.12.

### Experiment 6: Divergence parameters

Taking Experiment 5 as a starting point, the system parameters which control the mixture of the transfer components are scanned in sweeps. The transfer shape produces an effect that is reflected in different ways in the error, in the divergence of $X$ and $Y$ distributions, and also in the information distance between the two distributions. The combined results of Experiment 5 and 6 are shown in Figure 6.12. The main message to be taken from the plot is, that deterministic distortions do not affect the learnability of an inverse mapping indicated by the information distance in the top panel. A decreasing budget indicates the need for learning in general. The divergence corresponds to the amount of adaption necessary to compensate distortions which might affect to time needed for learning.
Putting these results into relation it can be seen that divergence and information distance measure two different kinds of things. The divergence measures the amount of mass that a full explanation would need to move. The information distance indicates, if moving mass from the current source will actually make a difference at the destination. The budget drives the necessity for better

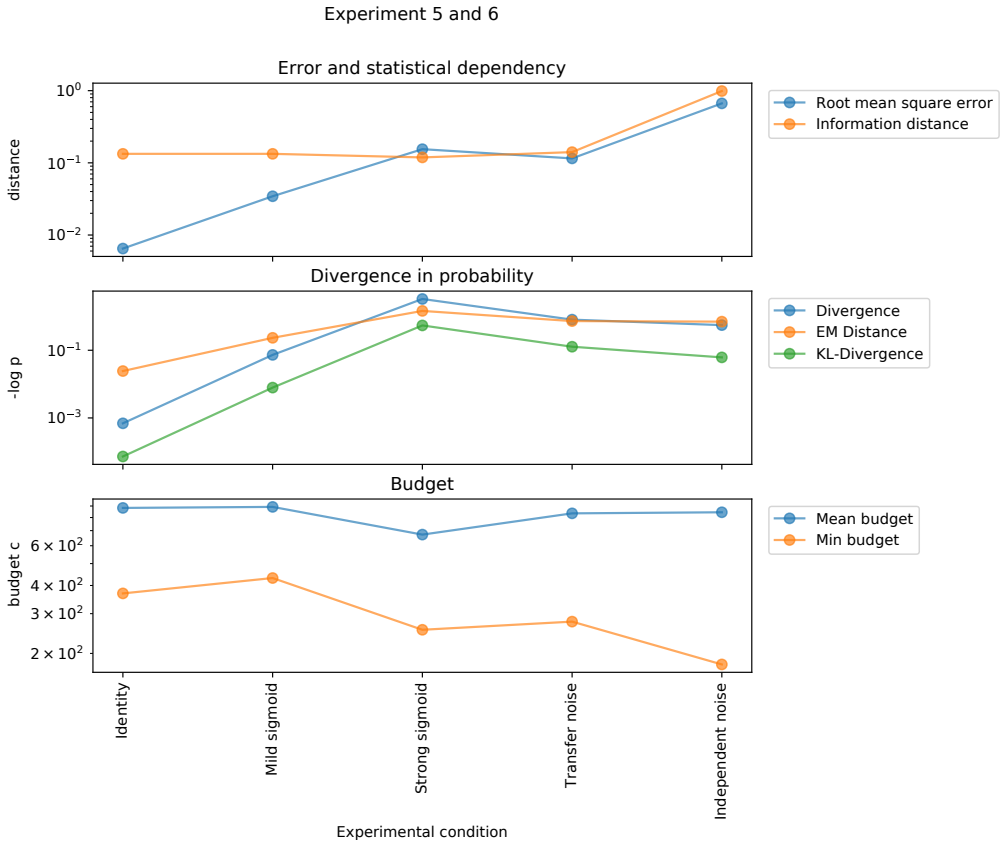



Figure 6.12.: This plot summarizes Experiment 5 and 6 with five different transfer conditions in total. The conditions are identity, mild sigmoid distortion, strong sigmoid distortion, transfer noise and independent noise. For each conditions the root mean square error, the normalized information distance, the chi square- and Kullback-Leibler divergences, the Earth mover's distance and the budget mean and minimum are shown. Results are averaged over 10 runs of 10000 time steps each. In the top plot, the error responds sensitively to deterministic distortions of the mapping whereas the information distance remains unaffected, which means learnability in principle. In the presence of external noise both curves respond strongly. The divergences in the center panel agree qualitatively with a pronounced peak for the large sigmoid distortion condition. Divergence corresponds to the amount of adaptation necessary to compensate distoortions. The bottom row contains the mean and minimum budget which can be interpreted as an increasing need to adapt with decreasing values.

explanation. These measures specify a range of what needs to be learned (budget) to what can be learned (information distance) using the error and divergence as learning signals. This excursion into divergence in probability served to frame the agent success in more differentiated terms. Quantitative divergence and dependency measures have been introduced and are proposed as a *composite* measure to quantify the required and obtainable adaptation from sensorimotor data. This can be compared with the effective adaptation accomplished by internal models, for example to predict survival probabilities.

## 6.5. Adaptive internal models

In this section, adaptive internal models are examined in detail. In mechanical terms, learning and adaption is accomplished by *fitting models* to sensorimotor *data*. The embedding of each model in the sensorimotor context is made explicit, and the relationship between this *contextual embedding* and the *functional role* that is acquired within the enclosing developmental model are defined and illustrated with examples.

### 6.5.1. Adaptive internal models

The idea of internal models in sensorimotor theory as it is currently used, is attributed to (Craik 1943), although several precursory formulations can be found in the literature, summarized in (Johnson-Laird 2004). Models inside other systems are conceptualized as functional building blocks enabling *anticipatory behaviour* (Rosen 2012), which is also an essential aspect in robotics (Winfield and V. V. Hafner 2018). An internal model enables an agent to cope with partially observable environments, by replacing missing observations with a corresponding prediction. Equivalently, a model can serve to fuse the prediction with the measurement and thus obtain an improved estimate of a given state variable (Thrun, Burgard, and Fox 2000). Through integration of multiple inputs, a model can accumulate the *synergistic information* contained in those channels when considered together (Wibral, Priesemann, et al. 2015).

The concept has been proven successful in more recent research. This is reflected in a large number of publications on internal model research. These include work on functional architectures (Daniel M Wolpert, Ghahramani, and Jordan 1995), (Daniel M Wolpert and M. Kawato 1998), (Haruno, Daniel M. Wolpert, and M. M. Kawato 2001) (Haruno, Daniel M Wolpert, and M. Kawato 2003), (Demiris and Khadhouri 2006), (Morse et al. 2010), perspectives from adaptive control theory (Tin and Poon 2005), neural correlates (Mischiati et al. 2015), and integrated approaches to development of cognition (Schillaci, V. V. Hafner, and Lara 2016) (Tononi, Sporns, and Edelman 1994).

The model configuration is either innate, learned, or acquired via a combination of both. In artifical systems required models are either provided as a prior to the agent or the agent is provided with means to adaptively learn such a model from sensorimotor observations. The priors can be encoded into the learning rules themselves, for example learning rate parameters, that express the inverse of the expected measurement noise. Long-term autonomy requires substantial capacity for learning models under varying pretexts. There might be more or less time available for exploration and learning, or a special instance of a task might require particular precision.

Given that an agents wants to restore the baseline performance by asking for closeness of the predictive and measurement distributions, it can do so by reducing divergence via the process of Equation 6.16, repeated here for quick reference, $\min_i d(\hat{\boldsymbol{s}}, \check{\boldsymbol{s}})$.

A functional requirement is implied, which is the ability to invert the accumulated perturbations of Equation 6.20. Starting with the motivation and substituting the generating operators A, B, E,

$$p(\hat{\boldsymbol{s}}) \overset{!}{\approx} p(\check{\boldsymbol{s}}) \tag{6.27}$$

$$\mathsf{A}(s) \overset{!}{\approx} \mathsf{E}(\mathsf{B}(\mathsf{A}(s))) \tag{6.28}$$

then the inversion task can be written as

$$\mathsf{A}(\underbrace{\mathsf{B}^{-1}(\mathsf{E}^{-1}(s))}_{\text{lumped model } M}) \approx \mathsf{E}(\mathsf{B}(\mathsf{A}(s))). \tag{6.29}$$

Using Equation 6.29, the agent is extended by inserting an adaptive model $M$, as in

$$\mathsf{A}(M(s)) \overset{!}{\approx} \mathsf{E}(\mathsf{B}(\mathsf{A}(s))). \tag{6.30}$$

The model $M$ can be considered from two perspectives. The external one describes, which variables are mapped to which input and output terminals, while the internal one is that of the learning algorithm inside of $M$, which is agnostic about the original context of the data. The *functional role* of the internal model emerges from the interaction of both parts and the data itself. Now the hypothesis is, that this largely determined by the context, and much less so by the learning algorithm or model type. In particular if the wiring is incorrect and acausal, the algorithm cannot be expected to repair that.
The next few experiments will demonstrate that the model will learn such an inverse map. The map accomplishes to take measurements back to the space of low-level predictions that originally caused them. The model is wired correctly by design, so that it necessarily learns the inverse prediction in a supervised learning setup. The wiring in general needs to be precise with respect to the sensorimotor configuration of time and modality, which is the topic of the Self-exploration chapter. It can be seen in the plots of the model's transfer function and divergence measures, that the motor distribution is correctly transformed by the model to generate the measurements according to the motivation of hitting goals.

### Experiment 8: Basic adaptive model

The configuration of Experiment 5 is extended with an adaptive model $M$. The model is a map taking measurements to their "causes" in motor space. This is achieved by assigning the measurements $\check{\boldsymbol{s}}$ to the model's inputs $X$ and predictions $\hat{\boldsymbol{s}}$ to the models targets $Y$. The experiment consists of an episode of 2000 steps, with a single model fitting and prediction step appended at the end of the episode. The resulting model's characteristics are shown as a sampled transfer function on top of the system's transfer function in Figure 6.13. The inverse relationship is reflected in symmetry of the curves around the identity diagonal of the diagram.

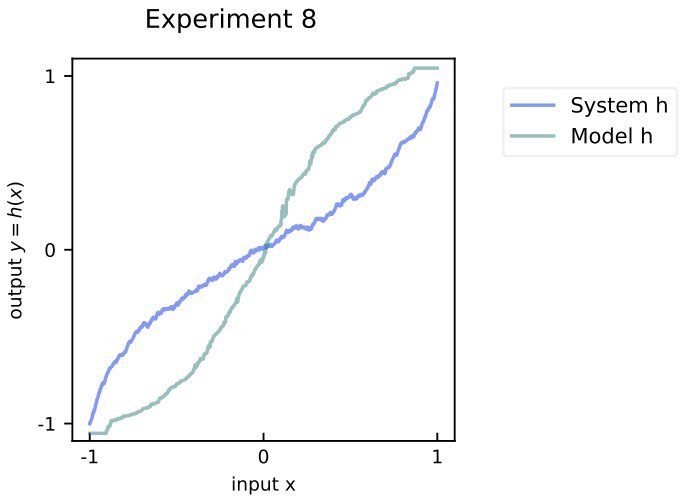



Figure 6.13.: Experiment 8: the system transfer function is shown in blue and the approximate inverse transfer function learned by the model is plotted as a green line. The symmetry of the curves across the diagonal confirms the inverse model function.

### Experiment 9: Online adaptation

This experiment is identical to Experiment 8 with the only difference, that the single batch fitting step is replaced with an online learning rule. The resulting transfer curve is plotted in the same way as before and shown in Figure 6.14. The fact that both figures agree apart from small random fluctuations comes from the interchangeability of batch- and online algorithms.

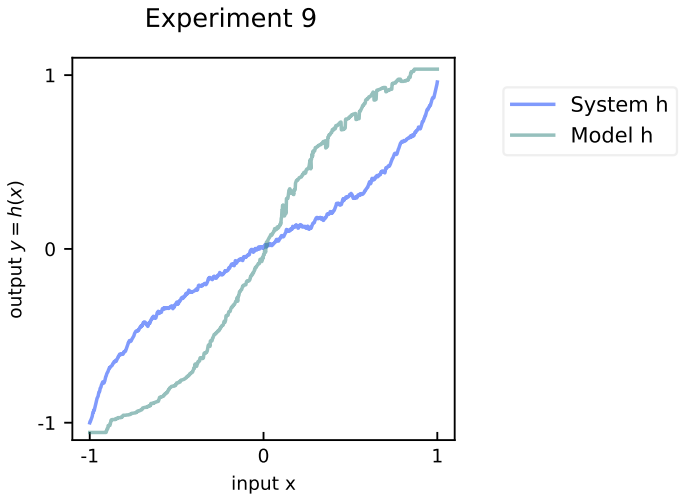



Figure 6.14.: Experiment 9: the system transfer function is shown in blue and the approximate inverse transfer function learned by the model is plotted as a green line. The picture is the same is in the previous experiment highlighting the interchangeability of batch and online updates.

### 6.5.2. Intermodal maps (s2s)

Some common functional roles are prediction of one modality from another one (intermodal map); prediction of some latent state with bottleneck constraints (encoder, information bottleneck); and prediction of a future state from past ones (forward model). Different types of stimuli caused by corresponding sensors are referred to as sensory *modalities*, for example seeing, hearing, smell and so on. Intermodal maps enable to predict the sensory state of one modality from the state of another. An important distinction is made between *proprio*-ceptive and *extero*-ceptive senses. The perception of internal states might also be referred to as interoception but should more accurately be included in the proprioception, which literally is the self-perception. Examples for proprioceptive quantities are self-generated forces, joint angles, and angular velocities, examples for exteroception are the tacile and visual senses, hearing, or chemical senses like taste and smell. Modalities are mapped to contiguous groups of columns $j$ of the sensorimotor data $\boldsymbol{S}_{i,j}$, and by combining them into the input of an adaptive predictor, synergistic information can in principle be obtained during learning (Cook and Bruck 2004; Williams and Beer 2010; Cook, Jug, et al. 2010; Wibral, Priesemann, et al. 2015), providing information about the environment not available through direct measurement. This is significant, as it not only allows an agent to infer a large amount of useful information about the environment (Philipona, J. K. O'Regan, and Nadal 2003; Terekhov and J. Kevin O'Regan 2016), but represents the entire basis of an agent's grounding (Poincaré 1905).

#### Experiment 10: Embodied agent

In this experiment an *embodied agent* is reobtained by putting the distortion model from the previous experiments back into the agent's body and environment. This establishes the first complete embodied agent which will be used and incrementally extended over the remaining sections of this chapter. The experiment appears to be identical to 6.4 but on close inspection, the computation graph is slightly different.
An example interpretation for this model is that of proprioceptive space. Proprioception means self-perception and refers to a sense of body configuration, conveyed via joint angle measurements. In this thesis, proprioception is used generally to refer to an agent's low-level motor space represented by the predictions, measurements, and other statistics of the corresponding variables. Other examples are an outgoing motor voltage (prediction) and measured rotation rate on a wheeled robot, or the predicted motor current and the measured torque on a joint on a torque-controlled robot.
If a robot actually is constructed in such a way, that there exists a sensor that measures something *physically* close to the action itself, it can be assumed that there will be a residual caused by microscopic but systematic divergence between actions and their corresponding measurements. The experiment shows an agent prepared to compensate for these deviations with adaptive inverse predictions. The resulting transfer ccurves are shown in Figure 6.15. The model's transfer curve in green is completely spurious and flat on average. The direct and model based measures given in Table 6.1 are identical, thus the model has no effect in this condition.

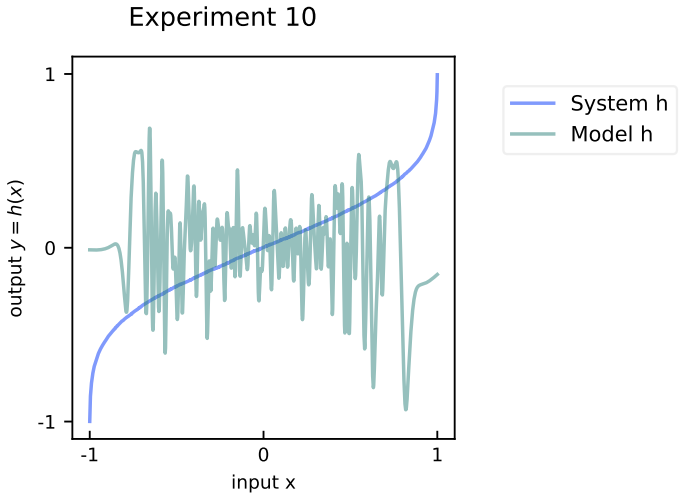



Figure 6.15.: Experiment 10: system transfer function (blue) and the inverse function learned by the model (green). Due to incorrect timing configuration the model learns a completely spurious mapping.

| Measure | direct | model-based |
|---|---|---|
| Information distance | 1.00 | 1.00 |
| Root mean square error | 0.65 | 0.65 |
| Divergence | 1.88 | 1.88 |

Table 6.1.: Experiment 10 tabular results showing that direct and model-based measures are the same and the model has no effect. This is due the broken temporal relation across the input and target.

Experiment 10 fails and the reason for the model's failure is, that the relation applied across the model inputs has become spread out in time. The inputs and targets of one model update step have become misaligned and the model effectively learns to predict a target that is statistically independent from the input. There are several simple design modifications that would fix this temporarily. Handling such *inevitable* delays more generally is a surprisingly complex issue. A more detailed discussion is deferred to the next chapter on self-exploration. For coherence of the presentation, a single step delay fix will be applied and the experiment be rerun to see more aspects of learning intermodal maps.

### Experiment 11: Embodied agent improved

This experiment fixes the delay problem of Experiment 10 by introducing a delay operator, which is configured with the *known* delay of one time step, and using the delayed prediction as the model's target input. This restores the proper temporal alignment of the input and target variables and a solid model can be acquired by the agent. The result is shown as transfer curves in Figure 6.16 where again the same picture of an inverse relationship as in Experiments 8 and 9 emerges, which is underlined by the model's errors in Table 6.2.

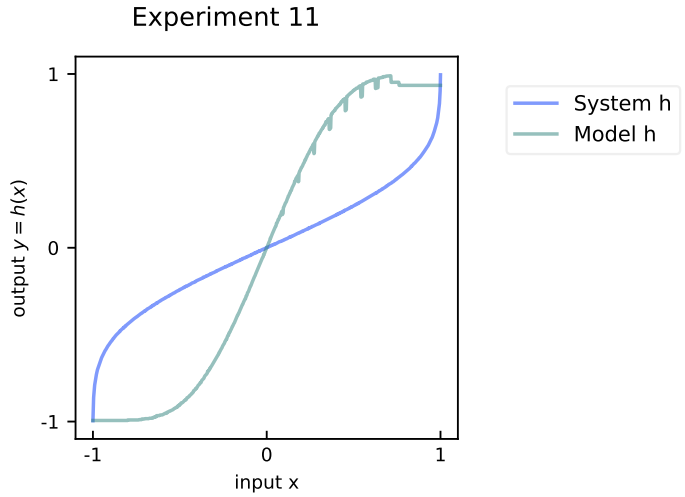



Figure 6.16.: Experiment 11: the picture is almost identical to Experiment 8 and 9 as the model is now trained with the original target delayed by one time step in comparison to Experiment 10. The effect can be seen in the restored symmetry of system and model transfer functions.

| Measure | direct | model-based |
|---|---|---|
| Information distance | 1.00 | 0.22 |
| Root mean square error | 0.67 | 0.04 |
| Divergence | 1.63 | 0.18 |

Table 6.2.: Experiment 11 tabular results. The direct measurements taken across system input and output are similar to previous figures, and the model based measures show a clear improvement in comparison.

An important question at this point is, how an agent gets to *know when to learn*. Obvious answers, besides being told from the outside, are the intensity of *reward*, and prediction *error* magnitudes. A reward tells the agent that past exploratory actions leading up to that reward can be reinforced and thus made more likely. The magnitude of prediction errors in turn tells the agent that there is a need for change through exploration and can be used to control exploration and exploitation. In the absence of any explicit reward, the consistent decrease of prediction error levels can itself be used as a reward.

The concept of *intrinsic motivation* is adopted for representing this type of processes. Several plausible models of motivation have been proposed, for example a motivational system (Otto E Rössler 1981), adaptive curiosity (Jürgen Schmidhuber 1991a; Jürgen Schmidhuber 1991b), homeokinesis (Der, Steinmetz, and Pasemann 1999), the autotelic principle (Steels 2004), learning progress (Kaplan and Oudeyer 2004), novelty search (Lehman and Stanley 2008; Juergen Schmidhuber 2009), intrinsic adaptive curiosity (Oudeyer, Kaplan, and V. V. Hafner 2007), and information driven approaches (Martius, Der, and Ay 2013; Salge, Glackin, and Polani 2013). All of them consist essentially of some kind of measure of recent behaviour, the current adaptation state and use that, to modulate exploration and exploitation. The minimum divergence motivation described in this chapter is identical to a minimum prediction error motivation, and

is consistent with these approaches. The same measures that have been used for analysis in the previous experiments are put into the sensorimotor loop in Experiment 12 and are used by the agent itself to modulate its sensorimotor model learning.

## Experiment 12: Embodied agent motivated

The final experiment of this chapter serves as a provisional connection to learning modulated by motivation. The connection is being made by combining an instantaneous error $e$ with low-pass filtered versions of itself integrated over different time spans. A primitive motivation $m$ controlling the learning can be derived from crossing points of different error integral. In the experiment the motivation is hardwired to activate a local model as soon as the accumulated error exceeds a given threshold, indicated by the yellow curve. Even when starting later in the episode, a correct model is acquired as shown in the usual transfer curve plot in Figure 6.17. and the error measures given in Table 6.3.

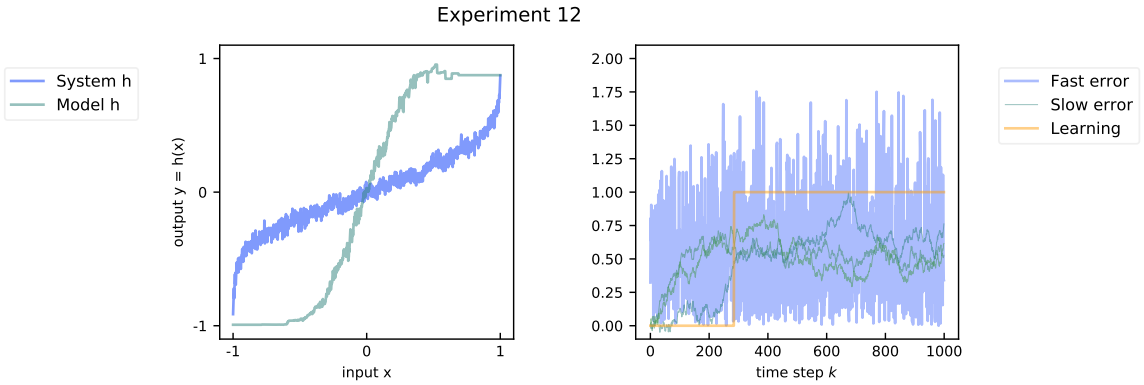



Figure 6.17.: Experiment 12 uses Experiment 11 as a basis. The difference here is that the activity of the adaptive internal model is modulated by a prediction error measure. The internal model only becomes active if the overall error exceeds a fixed threshold. This shows how the error signal can act as a minimal type of motivation signal which modulates and arbitrates among different model-based modes of behaviour.

| Measure | direct | model-based |
|---|---|---|
| Information distance | 1.00 | 0.68 |
| Root mean square error | 0.71 | 0.23 |
| Divergence | 1.70 | 0.43 |

Table 6.3.: Experiment 12 tabular results. Again, the model based error measures are improved with respect to the direct measures, indicating that a correct model was learned without precise control over when the learning occurs.

## 6.6. Results

With this, the basic framework at the level of development models for autonomously learning robots is complete for the scope of this thesis. Fundamental sensorimotor theory has been introduced along with a graphical language for specifying agent brains. These two have been brought together to experimentally validate their compatibility and explanatory completeness. This chapter provided the theoretical framework for sensorimotor experiments that was developed within the thesis. At the center of the framework is an *agent model* that facilitates design for learning autonomy by emphasizing an information based inside out view of agent activity. The framework has been implemented in a corresponding Python software library called *smp_graphs*. Based on this, a random strategy *baseline* for behaviour is defined, against which adaptive behaviours can be compared with several error measures. Finally, a blackbox definition of *adaptive internal models* is given and put to experimental use. The viability of the approach is shown with a set of experiments that provide the basis for the subsequent experiments done in the next two chapters.

# 7. Self-exploration

Starting from random strategies in the preceding chapter, *learning* was introduced as a way for an agent to compensate distorted access to the goal space. A succession of agents and their behaviour in isotropic environments was shown and incrementally modified, up to the point where the agent could learn to modify its action space to achieve better access to goal space. Some ad hoc *fixes* had to be made to the agent brain to obtain minimally working examples. *Self-exploration* is a process by which an agent can autonomously find solutions for the underlying problems in many different situations. These adaptations can be seen as *prerequisites* for skill acquisition, which are modelled here as fully adaptive phases in a developmental schedule.

## 7.1. Self and exploration

Exploration in everyday language means to go and visit uncharted places, and more metaphorically, to do something which has not been done before. This usually implies the hope of finding something not otherwise available. More generally, exploration emerges from the interaction of motivation and uncertainty and is the results of continuously sampling the motor uncertainty. The uniform random strategy is a plausible innate model (e.g. kinesis) and encodes maximum uncertainty about motor effects in the goal space. This garantuees that an exploration signal is always available to enable open-loop (aka off-policy) learning.

The self is subject of an ongoing general debate (Gallagher 2000), and increasingly so in computational accounts of cognition. As a working definition for the scope of the thesis, the self is all sensorimotor activity, which can be predicted at a critical average level of reliability. Reusing the diagram introduced earlier and shown again for quick reference in Figure 7.1, the agent self is graphically illustrated in Figure 7.2 as the outward extent of critical predictability. Proprioceptive space is the starting point and a candidate innate minimal self, with *proprio* meaning self, and it can be plausibly assumed to have low divergence and be well predictable, both in biology as well as in robots. Self-exploration on a kinematic system is used to illustrate this as an open-loop exploration baseline. Again, the limits of open-loop exploration are highlighted to motivate the hypothesis, that beyond a given threshold in body- and environmental complexity, exploration is necessarily incremental and closed-loop, to provide introspective measures fast enough for guiding the early learning transient. The hypothesis of self-exploration is that learning can be greatly accelerated if good priors for disciminating among self- and non-self regions of sensorimotor space can be found.

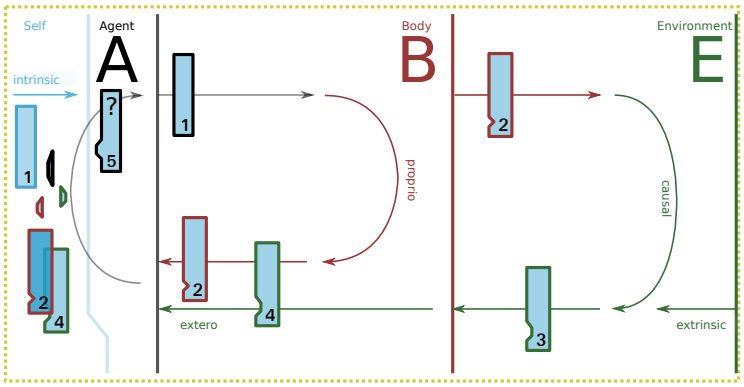



Figure 7.1.: This duplicates to the close up view of the cut across the sensorimotor information flow through the agent A, body B and environment E. There are two main cycles visible in the diagram. One is the proprioceptive cycle in grey / red arrows, which is close to agent by definition, and thus can also be expected to provide feedback much faster about the agent's actions' outcome. The other is the exteroceptive cycle, which is subject to increasingly indirect feedback paths, but providing valuable predictive information. The journey of a single information packet through is shown in numbered places along the the flow. The packet is duplicated and modified along the way, returning to the agent as measurements scattered over modality and time.

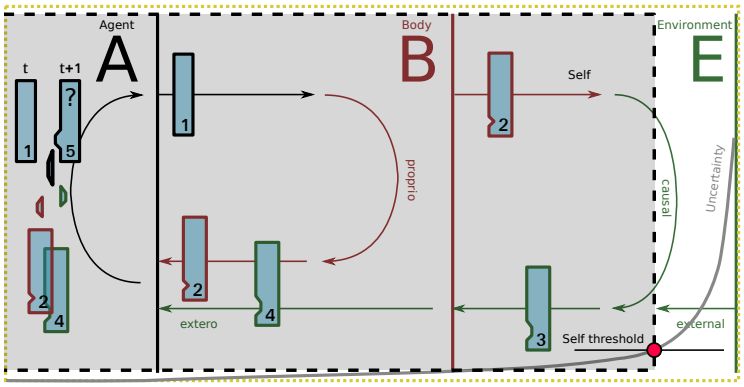



Figure 7.2.: Illustration of the *self* as the outward *extent* of critical predictability for a given agent A, shown as a shaded area over the previous diagram.

## 7.2. Tapping the sensorimotor trajectory

### 7.2.1. Introduction

Many theories of development contain the notion of brain modules acting as models of an agent's interaction with the world. These models can be used for example by the brain to evaluate possible actions in "imagined space" and the agent only commits to performing a promising action in physical space. The role of a theory on these models is to describe how precisely a sensorimotor model is learnt from experience and how it interacts with other existing models in a developmental context.
There are different types of such models. Machine learning (ML), for example, solves the problem of fitting a model to data in a problem independent form. The ML approach usually relies on a *preprocessing step* to transform the raw data into the required form. Using ML methods we can learn sensorimotor models of transitions in sensorimotor space up to a desired accuracy. This level of modelling provides the grounding in sensorimotor space. An important question is *how to map the raw sensorimotor data to sensorimotor training data* for realizing specific functions needed inside a developmental model.
The concept of *tapping* is adopted from signal processing where it is used to describe a filter as a weighted sum of delayed copies of a signal as shown in Figure 7.3. The simplest sensorimotor tapping then is just the same as a filter tapping, using past values of a single variable to predict a future value of the same variable. In realistic situations the number of past values can be numerous, include different modalities, and the linear filter is a general nonlinear function whose parameters are learned from data. This view allows us to discuss a wide range of issues in temporal learning. For example, concepts from developmental robotics, reinforcement learning, neuroscience, and information theory can be represented and compared by exposing relational properties independent of terminology.

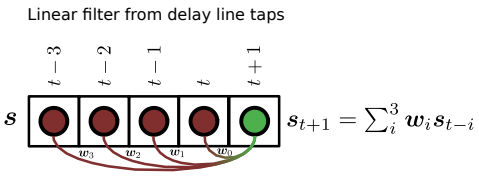



Figure 7.3.: This graphical representation of linear a filter uses successively delayed copies of an input $s$ to compute a prediction as a weighted sum of all copies. It provides the starting point for sensorimotor tappings.

### 7.2.2. Related work

A central concept in signal processing are linear filters. These were originally implemented as analog circuits using *delay lines* to store a finite amount of the signal's past values. In time-discrete implementations a filter's output is computed as a weighted sum over a finite number of past inputs. This is realized by *tapping* into fixed positions within a sliding window. Each tap is multiplied by a corresponding weight which together comprise the filter's coefficients. This provides the starting point for sensorimotor tappings. A filter can be seen as linear regression

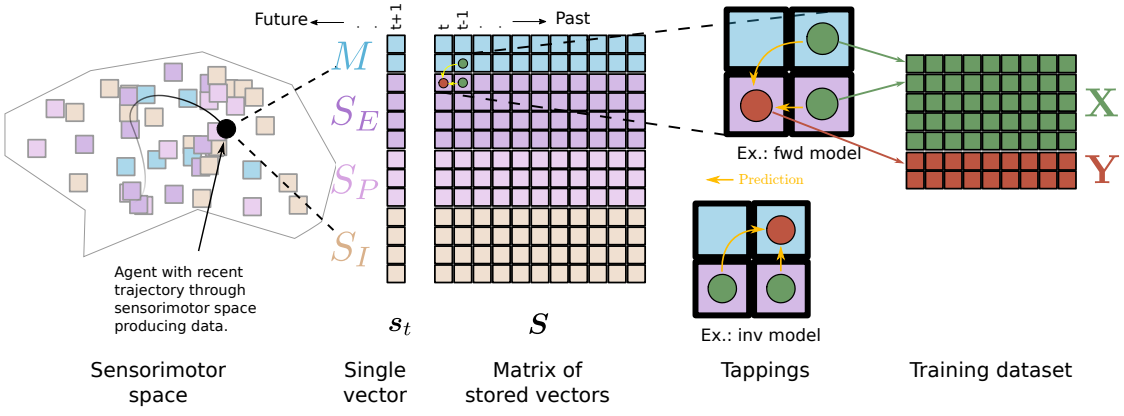



Figure 7.4.: The basic idea of tapping the sensorimotor trajectory. Concatenating the row vectors horizontally creates a matrix. The matrix inherits the row structure from the vector and represents time along the other axis.

and its coefficients can be learned with a least squares fit. This is known as an adaptive filter in signal processing and is the same as a linear adaptive forward model in a developmental robotics context.

The main techniques used for describing developmental models are plain text accounts, equations, and various types of block diagrams. Equations and diagrams are each highlighting different aspects of a model's function and behaviour. Equations are precise in representing functional dependencies including general temporal relations. Block diagrams emphasize which functions are used and which of those functions are interacting directly. None of them provides an intuitive representation of the global extent and the microstructure of interaction between variables for a given robot. This also means that reoccurring patterns of these properties and their systematic variation across different robots are hard to express.

More systematic graphical methods are the backup diagrams introduced in (Sutton and Andrew G. Barto 1998) and temporal probabilistic graphical models (Koller and Friedman 2009). Backup diagrams track how the instantaneous information is related to previous states and indicates how it is propagated back in time to update the relevant state in the agent's controller. These diagrams do not however differentiate sensory modalities very well. Probabilistic graphical models, especially dynamic bayesian networks, provide a natural complement to the current approach. Like recurrent neural networks, these models incorporate the problem of mapping input time and modality into the model state. In contrast, tappings aim at a decoupled representation of the input mapping and the model's state update.

Information theory can be used to quantify the amount of *shared* information among sensorimotor variables as shown in (Max Lungarella, Pegors, et al. 2005) or (Kaplan and V. V. Hafner 2006). This provides the empirical complement of tappings and can be used to obtain a tapping from data *prior* to training a model or to analyze a model's use of temporal information after training. A number of recent works have suggested *predictive information*, the amount of information shared between the past and the future of a random variable, as a measure for the amount of non-trivial information obtained from embodied interaction (W. Bialek and N. Tishby 1999). This

also highlights the importance of the agent's momentary temporal sensorimotor embedding. Internal modelling approaches in developmental robotics that use prediction learning are lacking a way to describe the interaction of the embedded sensorimotor models with the information provided by the enclosing developmental model in a general and systematic manner. This also holds for temporal difference learning in RL and correlational learning processes in neuroscience. Thus we see a definite need for an additional tool from which these fields, and maybe robotics and AI at large, might benefit. Our contribution besides the identification of this gap is a proposal for filling it.

### 7.2.3. Tappings

The sequence of steps necessary for going from sensorimotor space to the sensorimotor model input / output space are shown in the illustration in Figure 7.4 with enlarged views of two example tappings. A single sensory measurement at time $t$ is represented by a vector. The vector is composed of subparts that reflect the natural structure of the agent's *modalities* imposed by the sensors (e.g. vision or joint angles). The set of all possible vectors defines the agent's sensorimotor space. Measurement vector and sensorimotor space comprise the left part of the figure. The agent's internal time creates the temporal ordering of incoming measurements (Terekhov and J. Kevin O'Regan 2016), and storing them in this order forms a matrix. The matrix is shown in the center of the figure. It contains a numerical representation of the sequence of external states *as they are reflected* in sensorimotor space. An agent living in a partially observable world can benefit from extracting additional information from relations across time and modalities. To do this with memoryless models, the sensorimotor matrix has to be tapped using a context dependent pattern attached to the current time step with the data sliding along underneath. The patterns for a forward and an inverse model are shown close up. The locations of the nodes of the tapping indicate which relative time step and modalities are used to assemble a supervised training set. The node's colors indicate wether the datum is an input or a target.

#### Example

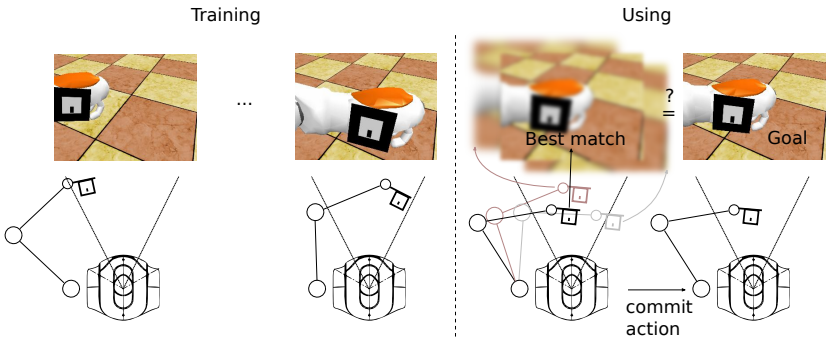



Figure 7.5.: On the left a Nao robot trains a model to predict visual consequences from joint angle configurations through sensorimotor exploration, right: the robot uses the model to find the best matching prediction and the associated action in the predictor's input.

Consider the example of a Nao robot bootstrapping the ability to move its hand to a given point in visual space shown in Figure 7.5. The agent creates an episode of data by exploring five random joint angles. For simplicity a kinematic arm is assumed so there is a delay of one time step between motor command and the corresponding measurement. Each momentary measurement consists of the *current* image, resulting from the previous command, and a *new* motor command about to be committed. In order to let the agent learn to predict the image in the next time step from the current command, an adaptive model is trained with commands as input and the image as target taken from different relative time steps as shown in Figure 7.6. The training set is created from the raw data by shifting the row of commands one time step to the right. The measurements in each column of the new matrix are now ordered by model update steps instead of sensorimotor time. A detailed tapping is shown Figure 7.7.

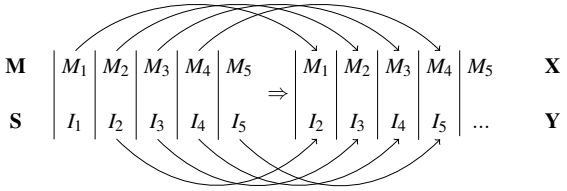



Figure 7.6.: An unrolled view of the repeated application of a tapping into sensorimotor data that the Nao agent uses for constructing the training data with inputs $\boldsymbol{X}$ and targets $\boldsymbol{Y}$.

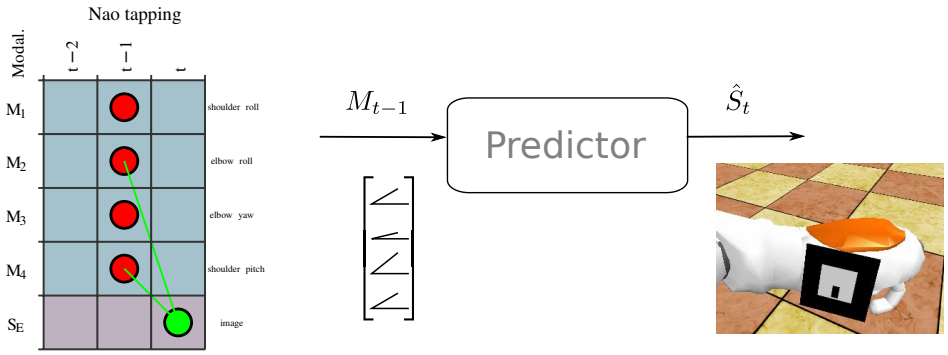
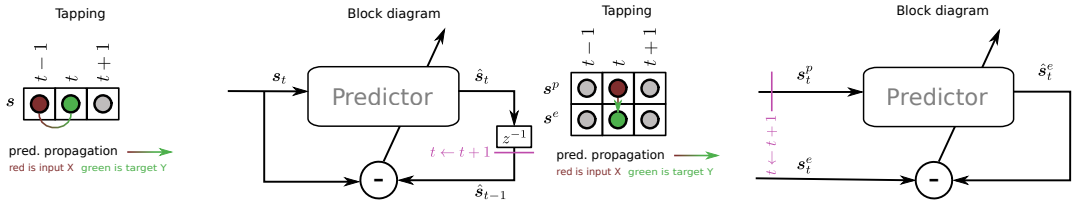



Figure 7.7.: Tapping for the Nao example with fully expanded motor signals and a corresponding block diagram.

### Tapping degrees of freedom

Tappings are specified relative to the current time $t = 0$, becoming positive in the future and negative into the past. This proposal only considers discrete time and equidistant sampling with a constant $\Delta t$. It makes sense to group variables in the matrix according to their modality such as as exteroceptive- (vision, hearing), proprioceptive- (motors, joint angles, forces), or *interoceptive* sensors. Interoceptive variables represent any intermediate stage of other concurrent computations in the agent's sensorimotor loop. A group whose elements all contribute to the same argument

of the target function, for example all pixels in an image, can be reduced to a single element in the graphical representation.
A common arrangement in a developmental model is to use a supervised learning algorithm because it can be trained effectively. A supervised training set consists of the input $\boldsymbol{X}$ and targets $\boldsymbol{Y}$ that constrain the functional relation $f(\boldsymbol{X}) = \boldsymbol{Y}$. The approximation task is to find parameters $\theta$ for the model $\hat{f}(\cdot, \theta) = \hat{\boldsymbol{Y}}$ such that $|\boldsymbol{Y} - \hat{\boldsymbol{Y}}|$ is minimized under a given loss. Prediction learning allows the agent to construct infinite supervised training data on the fly. Tappings can describe the necessary transformations independent of the learning algorithm. If $\boldsymbol{XY}$ is the full supervised training set, the tapping defines a map taking an $\boldsymbol{SMT}$ index set to an $\boldsymbol{XY}$ index set.
It can be immediately seen from the figures that a tapping is a directed graph on top of $\boldsymbol{SMT}$'s row and column indices. The graphical structure encodes the relation prescribed by the sensorimotor model's function inside the developmental model. In addition to the supervised learning case the graph can immediately be taken as dynamic Bayesian network graph connecting the current approach to a rich existing body of formalism and inference techniques.




Simple temporal predictor. Simple intermodal predictor.

Figure 7.8.: The two principal axes of association shown as tappings alongside with corresponding block diagrams. a) A simple temporal predictor, predicting the state one timestep ahead, and b) a simple intermodal predictor taking proprioceptive input to an exteroceptive prediction.

A single time step prediction problem requires a tapping from one time step to the next. Doing the same along modalities captures intermodal prediction, that is, predicting sensory consequences in one modality from the state of another modality. By adding joint angle sensors to the Nao agent, it could learn to predict the hand position (vision) from joint angles (proprioception) in the same time step.
The two dimensions of the sensorimotor data matrix result in two corresponding tappings resulting in a temporal predictor, shown in Figure 7.8, and an intermodal predictor shown in Figure 7.8. Sensorimotor models encode regularity in sensorimotor state transitions along these axes. Learning transitions along the normal forward flow of time results in a forward model. Forward models are central to the simulation theory of cognition, which states that an agent learning to approximate the forward transition rules to a sufficient degree of fidelity can use them to internally "simulate experience" (Hesslow 2012).
Rearranging the direction of prediction to go backwards in time creates an inverse model. This allows the model to predict (infer) causes from observed effects, which allows the agent to control and change its own state by directly predicting the causes of its desired state. This translates to predicting the actions that lead to a goal (M. Rolf and M. Asada 2015). Direct prediction imposes constraints on the learning algorithm. Generally the inverse of a function can be a correspondence,

requiring the learning algorithm to be able to represent this type relation.

**Summary**

To summarize this subsection we highlight the main features of tappings. They provide an information centric view on developmental models. This view is independent of particular learning algorithms, and it provides an upper bound[1] on the amount of explanation a model needs to accomplish. That bound is a reference for comparing different models in terms of the fraction of maximum explanation. Tappings facilitate the design of developmental models, algorithms and their implementations by highlighting regularities in the design space and being precise and explicit about time. Analysing two important model types and their tappings shows to what extent different functional roles are determined by the input / output relations, and the learning algorithm respectively. These features all contribute to facilitate systematic exploration of developmental models.

### 7.2.4. Basic tappings

In this subsection we explore tappings further by looking at some variations of the simple ones that came out of the previous subsection: multi step prediction, autoencoding, and autopredictive encoding. If the internal model is a feedforward map without internal memory the simple one time step predictor in Figure 7.8 cannot make use of additional information about the future that was presented more than one time step ago. The missing memory of the model can be replaced by using a *moving window* of size $k$ that augments the momentary model input by including all $k$ previous values of the variables[2]. Since tappings are moving windows, the multi time step tapping shown in Figure 7.9 is almost trivial, the window size being equal to the number of input taps spread uniformly into the past. Iterative predictions in extended forward simulations demand better model accuracy. A reasonable shortcut towards more accuracy is to improve the prediction by imposing a long-term consistency constraint by extending the target tapping into the future (using buffering in closed-loop learning).
A special case of a predictor is the autoencoder. Its tapping is shown on the left in Figure 7.10. Its target output is the same as its input. In terms of the $\boldsymbol{XY}$ formulation with $\boldsymbol{X} = \boldsymbol{Y}$, the autoencoder could only consist of wires. The added value of an autoencoder comes exclusively from constraints on the intermediate representation. Like prediction learning, autoencoding is an unsupervised learning technique built with supervised learning. If we look at the tapping we see that the information of each single variable on the input is distributed to all other variables on the output. By a simple change of the tapping we easily obtain an autopredictive encoder (APE) as the result of pulling the autoencoder's input and output taps one time step apart. The autopredictive encoder is not an established term but multiple proposals for such architectures have in fact been made (Michalski, Memisevic, and K. Konda 2014; Patraucean, Handa, and Cipolla 2015; Copete, Nagai, and Minoru Asada 2016). Applying the prediction constraint on the model has been shown to increase the task-independence of latent space representations by

[1] the joint entropy of all sensorimotor variables
[2] The moving window technique is alternatively known as moving average model, time delay neural network, delay-embedding or method of delays

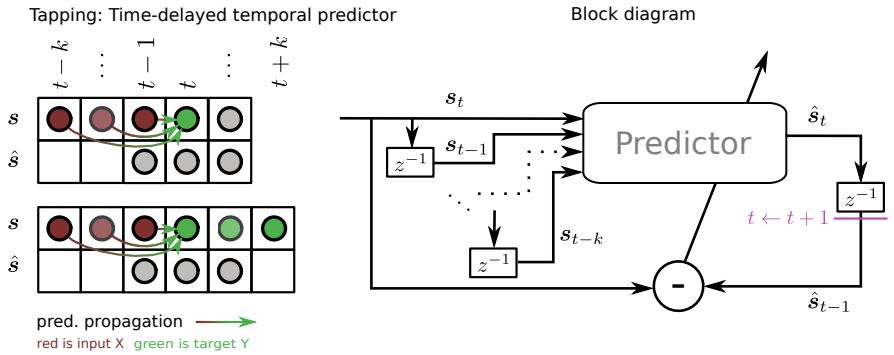



Figure 7.9.: The multi step predictor using a window on $k$ past values as instantaneous input and, in the fully symmetric case a window on $k-1$ additional future values as the target. The time indexing has been omitted for simplicity.

(Lotter et al. 2015, (Lotter, Kreiman, and Cox 2015)). In the tapping we see immediately that the prediction constraint encourages the model to represent the rules of change in the hidden space. The APE tapping is shown in Figure 7.10.

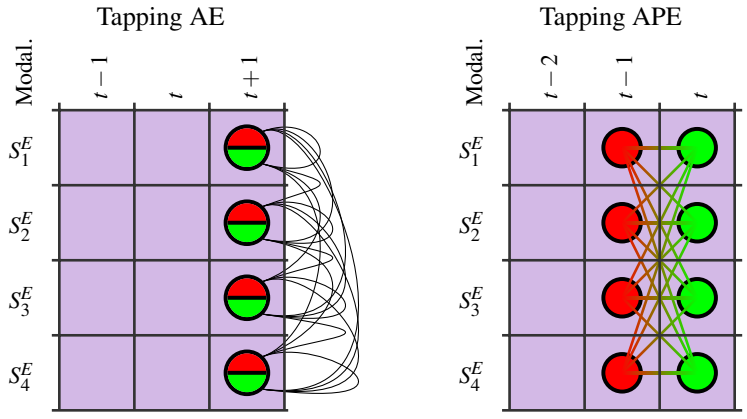



Figure 7.10.: Autoencoder (left) and autopredictive encoder (right). The AE's tapping is special because input and target coincide. Pulling the input and source apart over one timestep difference produces the autopredictive encoder. The prediction prior imposes additional structure on the hidden representation.

### 7.2.5. Application areas

Internal modelling (Craik 1943) is an important concept used in developmental robotics (Daniel M Wolpert and M. Kawato 1998; Demiris and Khadhouri 2006; Schillaci, V. V. Hafner, and Lara 2016). An underlying driving hypothesis is that predictive models enable *anticipatory* behaviour (Rosen 2012) which is more powerful than purely reactive behaviour. From the developmental perspective this implies that some functions of a developmental model must be provided by adaptive models of the sensorimotor dynamics. Two basic functional types of internal models,

forward and inverse ones, have already been introduced as examples in Figure 7.4 and are shown again as a pair of tappings in Figure 7.11. This highlights the rearrangement of the direction of prediction without a change of variables. Exploitation of adaptive models has also been described above indicating different ways of predicting and evaluating future options with forward models, or directly inferring actions with inverse models.

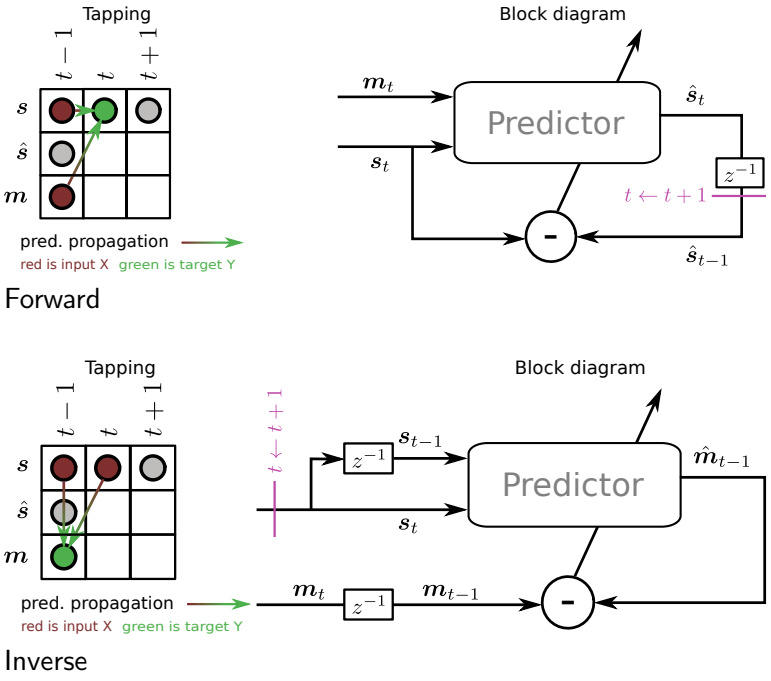



Figure 7.11.: Tapping a single time step forward- and inverse model pair. The model's functions are determined by different relations over the same set of variables.

A popular method in reinforcement learning is temporal difference learning. Temporal difference learning is a family of algorithms to approximate a prediction target with a recurrent estimate. The usual target is a *value function* which maps actions to a value. The estimate is bootstrapped by minimizing the moment-to-moment value prediction error, which is ultimately grounded in a primary reward signal. There exists extensive theory in RL that deals with the problem of integrating task-relevant information that is spread out in time, with two fundamental concepts being involved. The first one is that of *multistep methods* which take care of consequences escaping into the future. The second one are *eligibility traces* which capture causes vanishing into the past. Taken together they solve the general delayed reward problem. Depending on the parameters a corresponding tapping will be similar to the multi step predictor.
The importance of features and modalities and the information contained in their mutual relations is less developed. The concepts used in reinforcement learning can easily be remapped to internal modelling terms and vice versa, making tappings immediately applicable to temporal difference learning problems. Looking at three basic temporal difference learning algorithms, TD(0), Q-Learning and SARSA, it can be seen that they all approximate a target by updating from a one

time step difference. TD(0)'s target is a state value function $v$ while for Q-learning and SARSA it is a state-action value function $q$ (Sutton and Andrew G. Barto 1998). The update rules all follow the same general form of

$$\Delta v = \alpha(R_t + \gamma v(S')_t)$$

and the corresponding tappings are shown in Figure 7.12. Comparing these with the internal model tappings we see that temporal difference learning corresponds with prediction learning and that the value function is a forward model allowing us to reframe RL problems as developmental prediction learning ones and the other way round. The $\lambda = 0$ case is shown here to correspond to a single time step tapping but the proportional increase in tapping length with increasing $\lambda$ should be obvious.

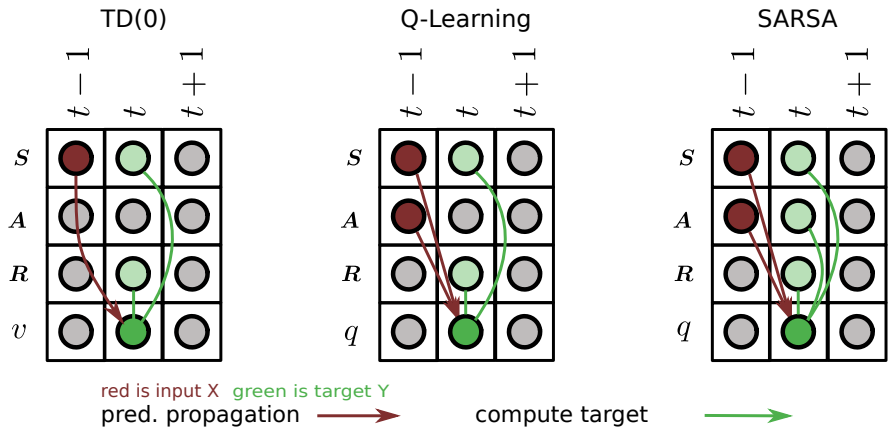



Figure 7.12.: Tapping temporal difference learning algorithms.

Neuroscience provides several models that link computational and neurobiological accounts of associative learning and reinforcement learning. The Rescorla-Wagner rule (Rescorla and Wagner 1972) is one example. It is a model of classical conditioning and describes how an association is learned across two modalities, the unconditioned (US) and the conditioned stimulus (CS), which occur at different times. Another example is the reward prediction error hypothesis of dopamine (Wolfram Schultz, Dayan, and Montague 1997; Dayan 2002; Niv 2009) which provides a physiological mechanism in support of computational descriptions of reinforcement. Low-level models of neural adaptation like spike-time dependent plasticity (STDP) (Gerstner et al. 1996; Markram et al. 1997) are characterized by a local window of interaction on a microscopic time scale. STDP itself is not a model for learning delays but an even lower level mechanism for reinforcing or weakening the association of pre- and post-synaptic events based on the local window prior. It can of course be used indirectly to extract sensorimotor delay information. Tappings apply without modification to all these different levels of modeling as shown exemplarily for the conditioning case in Figure 7.13.

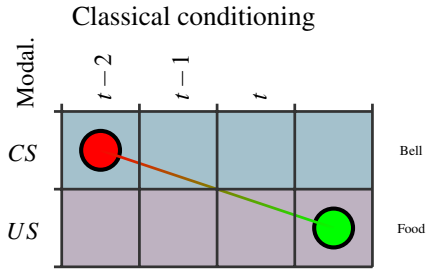



Figure 7.13.: Model of classical conditioning: it explains the prediction of the unconditioned stimulus (US) from a stimulus occurring earlier in time, the conditioned stimulus (CS). The predictive association of stimuli across time is precisely the process of conditioning. This highlights again that the difference to a forward model or a value prediction is only in the terminology and not in the structure of the association problem.

### 7.2.6. Discussion

During this presentation of tappings, a few additional issues came up that still need to be discussed. Models with memory like recurrent neural networks or dynamic Bayesian networks need special consideration with respect to tappings. Such models naturally retain an internal memory of past input values. Because of this, they do not need explicit memory in their inputs and in theory only need to tap across one time step. They are building up an implicit tapping as part of their learning while tappings aim at an representation of specific memory needs for a given learning task. Measuring the information flow across the model inputs and outputs after training with quantitative (Max Lungarella, Pegors, et al. 2005) or relational techniques (Williams and Beer 2010) should result in an effective tapping that could be used for comparison with prior tappings or interpreted as a way of learning them.
The memory issue is an example of a more general aspect about tappings. The current proposal disregards details about the learning algorithm used at the level of sensorimotor models. It is argued that this is in fact an advantage and necessary for wider comparison of models. The same is evident in the case of inverse problems where the learning of correspondences instead of functions needs to be considered. It remains to be shown how these properties could be integrated and represented in a tapping.

### 7.2.7. Conclusion

Tappings, a novel concept in sensorimotor theory for design and analysis of adaptive models in a developmental context was introduced. Tappings came out of a need for capturing the detailed embedding of learning machines in the temporal and modal context of raw sensorimotor trajectories. Tappings create a particular view on the interaction between the embodiment and the functional requirements of behaviour that can help to better understand developmental learning processes, and make sensorimotor learning more efficient. They can systematically describe the relationship between supervised learning and developmental models. By ignoring computational details the tapping view highlights the information flow across models and using that we can compare a large range of models that cannot easily be compared otherwise. We showed the

structural similarity of prediction learning in the developmental context and temporal difference learning in RL.

## 7.3. Quantifying tappings

The tapping concept introduced in section 7.2 is entirely abstract, and does not include any constructive description so far. In this section, it is shown how the concept is supported by information theoretic measures, and how these methods can be turned into a learning algorithm for a quantitative tapping (qtap) in a particular sensorimotor context.

### 7.3.1. Basics

In current approaches focussing on the self-organisation of behaviour, information based measures take the role of the search objective to be maximised. Here, information measures are used to support the actual learning performance of *any* model, regardless of the objective. The main idea is to *scan* over a set of pairwise combinations of variables chosen from the sensorimotor data matrix $\boldsymbol{S}$. Evaluating a dependence measure $d(\cdot,\cdot)$ on each pair results in a matrix $\mathbf{DEP}^{t_{\text{scan}} \times s_{\text{scan}}}$ of dependence estimates for each pair. The shape of the matrix corresponds with the configured extent of the scan over channels and time, $s_{\text{scan}}$, and $t_{\text{scan}}$ resp.. Scanning allows to sieve the variables for mutual dependency indicating their predictive relevance. From that, a corresponding graph can be constructed that *embeds* the adaptive model terminals in the sensorimotor data stream.
The scan result can formally be regarded as a filter, or a fractional integration operator, and when thresholded as an embedding. An embedding is a structure preserving map that creates new data points from existing ones with index operations only. The embedding can be combined with transformation by a map $\phi(X)$. The process of summing over selected axes of the embedding space can be interpreted as fractional integration, where the exponent of the integration operator controls the measure's locality. A well known signal analysis method is the short time Fourier transform (STFT), which is also a scanning method. The STFT is defined as a scan over an input $X$ using a window-size $k$ and a shift $n$ as parameters that control the resolution and locality of the overall result, called a spectrogram. In the infoscan method the sliding window part is identical to the STFT while the measure applied to each window is changed from the Fourier transform similarity measure to mutual information (MI) or conditional mutual information (CMI).

#### Experiment 13: Sensorimotor lag

In this experiment the action consists of uniform noise sampled at intervals of 5 steps for a duration of 20 steps. The robot has an inherent delay from motor input to sensor feedback of 2 timesteps. An agent does not know the timing parameters a priori for all bodies, environments or tasks. The agent could be supplied with all past and multimodal information but quite often, the relevant variables are sparsely distributed within any contiguous submatrix of SMT. Knowing the sites of relevant variables greatly increases the speed of learning. In this case, the sensor reponse is linear in the motor input so the temporal offset can easily be found with cross-correlation methods. The results are shown in Figure 7.14.

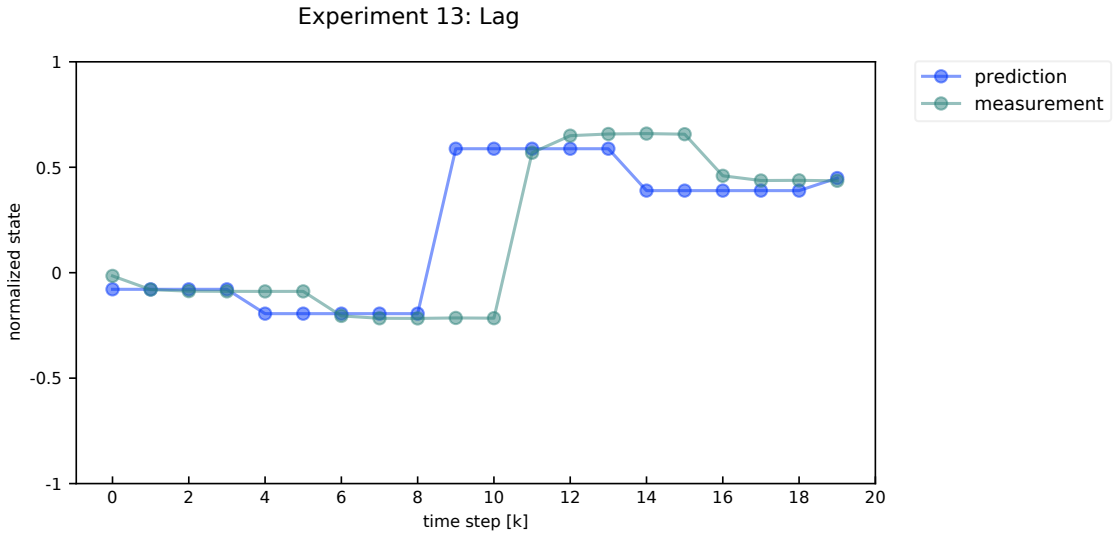



Figure 7.14.: Experiment 13 illustrates the temporal offset lag of a measurement $\check{s}$ drawn in green with respect to the corresponding motor prediction $\hat{s}$ shown in blue. The green curve follows the shape of the blue curve with a constant offset of two timesteps. The curves do not overlap precisely because of noise and small distortions that are present in the system.

Cross-correlation is a measure of vector similarity defined as the sum of the element-wise products. It is commonly used for signal matching and pattern search tasks. The question about transmission delays within the sensorimotor loop can in many cases be answered by searching for the motor pattern in the resulting sensor pattern. In Figure 7.14 the relationship between the motor prediction $\hat{s}$ and sensor measurement $\check{s}$ is approximately linear and the temporal offset can be read off the plot as two time steps. Scanning for the cross-correlation peaks for pairs of motor and sensor variables enables an agent to determine the delay and to adjust its tapping of the incoming data stream.

### Experiment 14: Lag from cross-correlation

The same configuration as in the previous experiment is now run for a full episode of 2000 time steps. A cross-correlation scan is performed on the motor predictions $\hat{s}$ versus the sensor measurements $\check{s}$. The scan consists of computing the Pearson correlation coefficients of $\hat{s}$ and $\hat{s}_i$, for a given scanning interval, which here is $i \in [-10, 0]$. The output of the scan, which is also called the cross-correlation function, correctly determines a delay of 2 time steps as the maximum correlation over all possible shifts. The result shown in Figure 7.15 is the multivariate sensorimotor timeseries as in Figure 7.14 but extended over 2000 time steps. It can be seen that the green measurement curve matches and covers the blue prediction curve but time shift is not visible anymore at the resolution of the plot. The cross-correlation scan result in Figure 7.16 shows a clear peak at a relative shift of two time steps. Since the measurement is shifted in reference to the motor time, the time indices are negative in the plot.

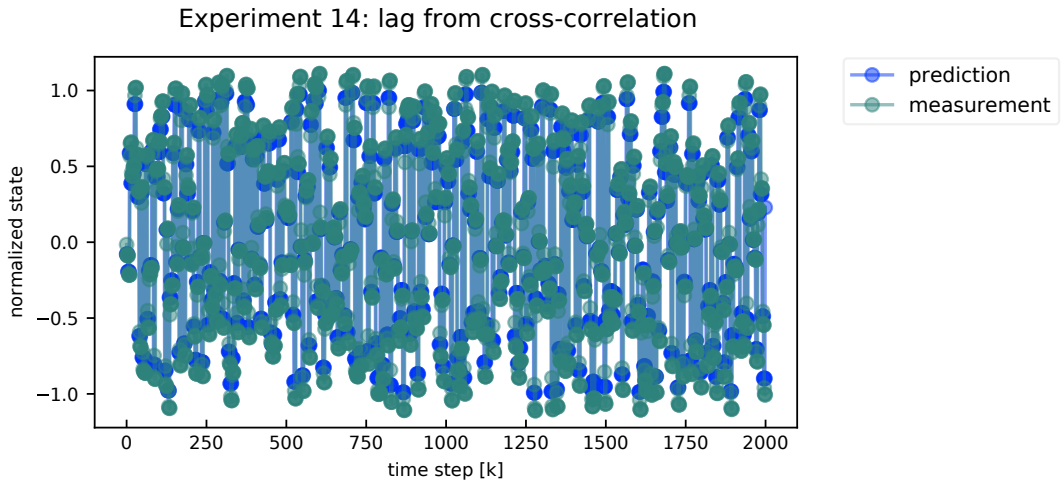



Figure 7.15.: The sensorimotor timeseries of Experiment 14 showing the sensor measurement $\check{s}$ in green on top of the motor activity $\hat{s}$ in blue. The fact that green dominates the picture means that the two variables are closely matching in value. The time shift is not visible anymore at the resolution of the plot but will be highlighted again in the cross-correlation plot below.

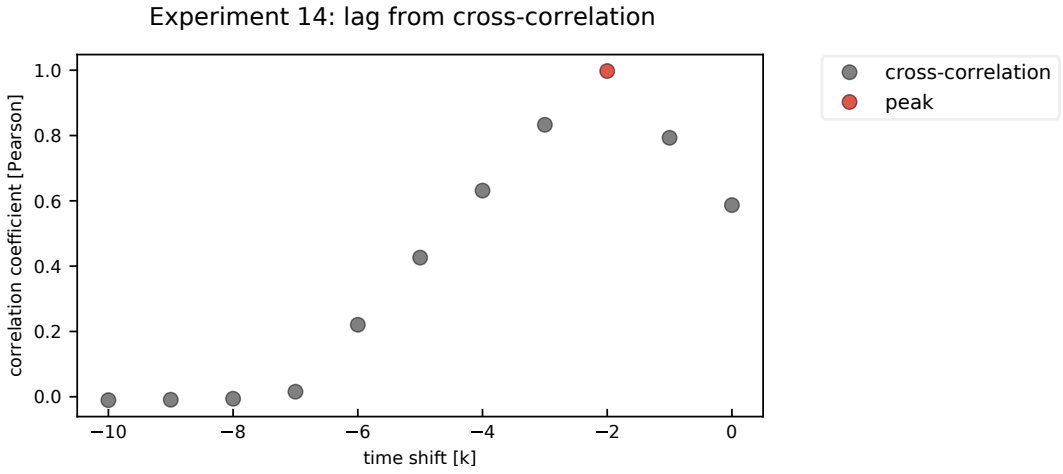



Figure 7.16.: Cross-correlation scan of motor prediction $\hat{s}$ and sensor measurement $\check{s}_i$ with the measurement shifted by $i \in [-10, 0]$ and indicating a peak at offset $i = -2$. This is equal to the ground truth motor to sensor lag configured in the experiment.

### 7.3.2. Quantified information

The following sections require to briefly introduce *information theory*. This theory (Cover and Thomas 2006) provides a framework for quantifying information. It was conceived in the 1940's by Shannon, Wiener, and Kolmogorov as a mathematical theory of information transmission, transformation, and storage. Since then, many extended measures have been proposed for quantifying interactions in complex systems, like sensorimotor networks.
The basic unit of information theory is *information* $\mathbf{i}$, also called *surprise*. It is defined over (the distribution of a) random variable $X$ and a singular event $X = x_k$ with a probability of occurence $p_k$. The amount of information gained through the observation of $X = x_k$ is

$$\mathbf{i}(x_k) \quad = \quad \log\left(\frac{1}{p_k}\right) = -\log p_k \tag{7.1}$$

The Shannon entropy $H(X)$ of a random variable $X$ is defined as the expected information, that is, the average amount of information per observation,

$$H(X) = \mathcal{E}[\mathbf{i}(x_k)] = \sum_k p_k \mathbf{i}(x_k) = -\sum_k p_k \log p_k \tag{7.2}$$

This quantity measures the amount of *uncertainty* inherent in a given distribution of a single (univariate) random variable. The *joint entropy* of two (or more) random variables $X$ and $Y$ measures the combined uncertainty within their joint distribution and is written as

$$H(X,Y) = -\sum_x \sum_y p(x,y) \log p(x,y) \tag{7.3}$$

for the bivariate case and can be extended to the multivariate case. The joint entropy is equal to the sum of the individual entropies if, and only if, the variables are statistically independent. Any degree of mutual dependency will register as a reduction in the joint entropy compared to the sum of component entropies. Such a dependency produces some amount of information $R$ that is *shared* among the two variables, also known as *redundant* information. The quantity can be computed as the difference between the sum of components and the joint entropy. Equivalently, the *conditional entropy* of $X$ given $Y$ can be computed, which is the uncertainty remaining about $X$ when $Y$ is known already, and it is given by

$$H(X|Y) = -\sum_{x \in \alpha_x} \sum_{y \in \alpha_y} p(x,y) \log p(x|y) \tag{7.4}$$

The previous three quantities are related by the following *chain rule*

$$H(X,Y) = H(X) + H(Y|X) \tag{7.5}$$

With this relation, many interesting measures can be composed. Most importantly, the mutual information is a measure of the statistical dependence of two variables, equal to the amount of shared information, and given by

$$m_I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$$

.
The complementary measure to mutual information is the *information distance*, introduced by (Crutchfield 1990) and written

$$d_I(X,Y) = H(X,Y) - MI(X;Y)$$

Thus mutual or shared information can also be thought of as information closeness. Both can be normalized by the joint entropy to obtain values in the interval $[0,1]$ which will be called $\overline{m_I}$ and $\overline{d_I}$.
If the random variables $X, Y$ are taken to be coding for the past and the future of a single process and are renamed to $X_-, X_+$ accordingly, the mutual information $m_I(X_-; X_+)$ is called the *predictive information* (PI) (W. Bialek and N. Tishby 1999). The PI has previously been called *effective measure complexity* (Grassberger 1986), and *excess entropy* (Crutchfield and Feldman 2003). Empirical estimates of the PI can computed by particular scan configurations.
The mutual information can also be extended to a conditional mutual information $m_I(X;Y|Z)$. A property of MI is, that it measures statistical dependence regardless of the direction of coupling, which is required when considering causality. A condition can be used to account for candidate contributions to a joint entropy $H(X,Y)$ to improve the causality estimate. If the MI of a motor and sensor variable is conditioned on the entire motor variable's past, any remaining entropy must clearly have been transferred in the current time slice. This is the transfer entropy (Schreiber 2000), written $TE(Y,X) = m_I(Y_t + 1; X_t^l | Y_t^k)$. In this context, behaviour can be seen as being generated by a network of interaction among data cells within the sensorimotor data $\boldsymbol{S}$, the unrolled sensorimotor loop. In summary, the conditional mutual information (CMI) is going to be used in what follows to measure dependencies among sensorimotor variables.

### 7.3.3. Nonlinearity

The next few experiments serve to show the effectiveness of quantified information in determining dependencies among variables in comparison with the well known cross-correlation analysis. In sequence the experiments will step through nonlinear, integrating, and multimodal relationships among sensorimotor variables.

#### Experiment 15: Nonlinear dependency

When Experiment 14 is repeated on a system with a nonlinear relationship among motor and sensor values, for example $s = \cos(\hat{s})$, the cross-correlation method fails. This is because the correlation

measure only captures linear relationships. The dependency information can be restored by using the mutual information instead of cross-correlation as the point-wise dependency measure. In Figure 7.17 the familiar sensorimotor timeseries is shown. In this plot it can be seen immediately that the green curve does not match the blue curve very well anymore. In Figure 7.18 two scans are shown, the cross-correlation in gray and the mutual information in red. The cross-correlation is flat and the maxima are spurious whereas the mutual information exhibits a clear peak at a time shift of two time steps, which is the ground truth configuration.

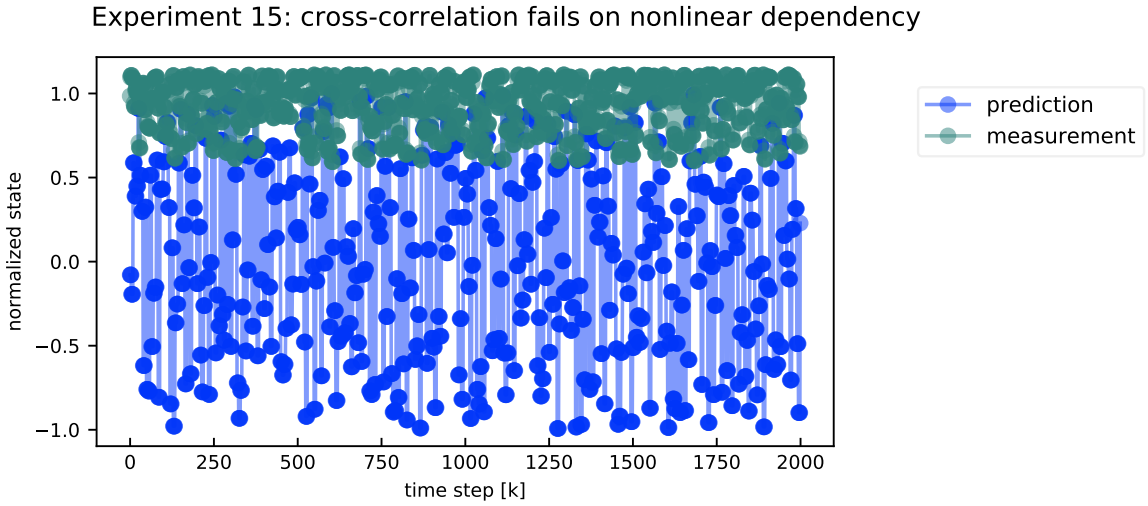



Figure 7.17.: Timeseries of the motor values $\hat{s}$ in blue and the sensor values $\check{s}$ in green. The motor-sensor relationship of this system (a joint angle controlled cartesian end-effector coordinate), is still systematic, but not linear anymore. The green sensor responses can be seen lumping together in the positive half-plane.

### 7.3.4. Integration

A sensorimotor variable can be an approximately integrated version of another variable, for example a velocity measure corresponding to an integrated acceleration, or an angle corresponding to and integrated angular velocity. The mutual information across such a pair of variables is measured in the next experiment.

#### Experiment 16: Integral relationship

The system used in Experiment 15 is extended further by an additional *order*. This means, the dimension of the primary motor variable at order 0 is kept the same but an additional variable is introduced into the system state at order 1, which is computed by integrating the order 0 variable. A simple interpretation is the relation of acceleration and velocity. Also, the nonlinear functional relationship between motor and sensor values at order 0 is being kept. The coupling

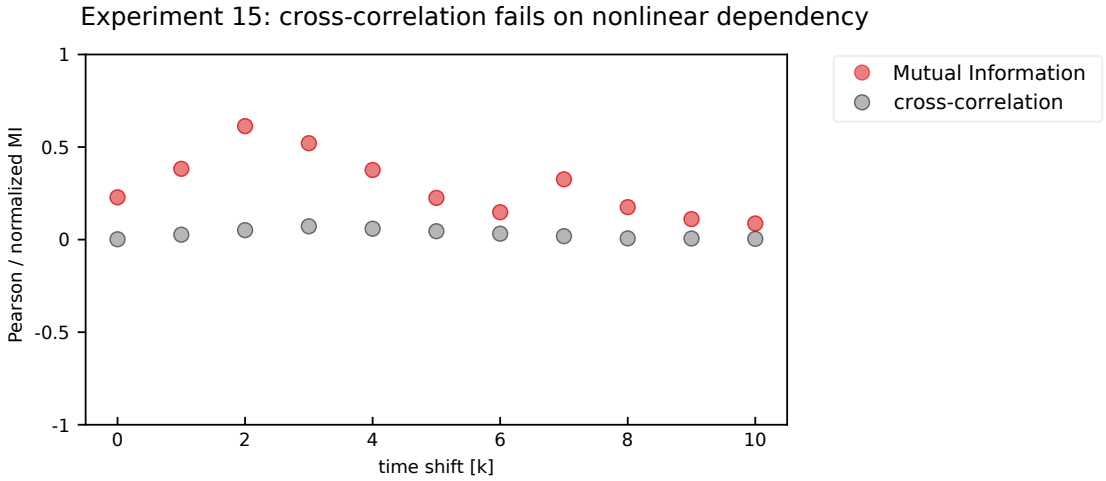



Figure 7.18.: Cross-correlation scan in grey and a Mutual Information scan in red for identical shifts. Normalized correlation coefficients take on values in the interval $[-1, 1]$. The normalized mutual information has a range of $[0, 1]$. Cross-correlation does not pick up the systematic dependence of $\check{s}$ on $\hat{s}$, which is indicated by correlation coefficients close to zero. In addition, the peaks of the cross-correlation are spurious. The mutual information restores the capability of cross-correlation in the linear case as can be seen as a clear peak of the information dependency at a relative shift of two time steps.

between action and effect is set to a lag of two time steps as in the previous experiments. The raw timeseries of the full system state is shown in Figure 7.19 with the motor and proprioceptive signal (order 0) shown in blue and green, and the velocity (order 1) shown in bright green. The presence of three different colorbands in this plot is a visual indicator of differing distributions of the three variables.
Velocity is computed by integrating the acceleration with a dissipative term modelling friction. Thus, the velocity cannot grow without bounds and saturates close to a value of 0.55. The scan results are shown in Figure 7.20. Four pairwise scans are performed in total on the pairs $(m_0, s_1)$, and $(s_1, s_1)$ using the cross-correlation and the mutual information measures. The first pair is the motor signal (order 0) and the velocity sensor (order 1), the second one is the self-pair of the velocity sensor. The system is designed so that the information in the velocity is determined both by a cross-modal action and an intrinsic memory. The memory is caused by inertia in this case. Cross-correlation fails again to detect the nonlinear and integral relationship between action and velocity, which mutual information is able to capture. The scan is performed over a range of 40 timesteps and the results show that temporal dependencies are close to the current time step and compactly distributed. For the given window size, the dependency measure over time converges, indicated by values close to zero for all measures starting from ten time steps into the past.

### 7.3.5. Modalities

Different perceptual modalities are usually related through a mixture of nonlinearity and order differences. The following experiments are examples of information scans applied to more complex

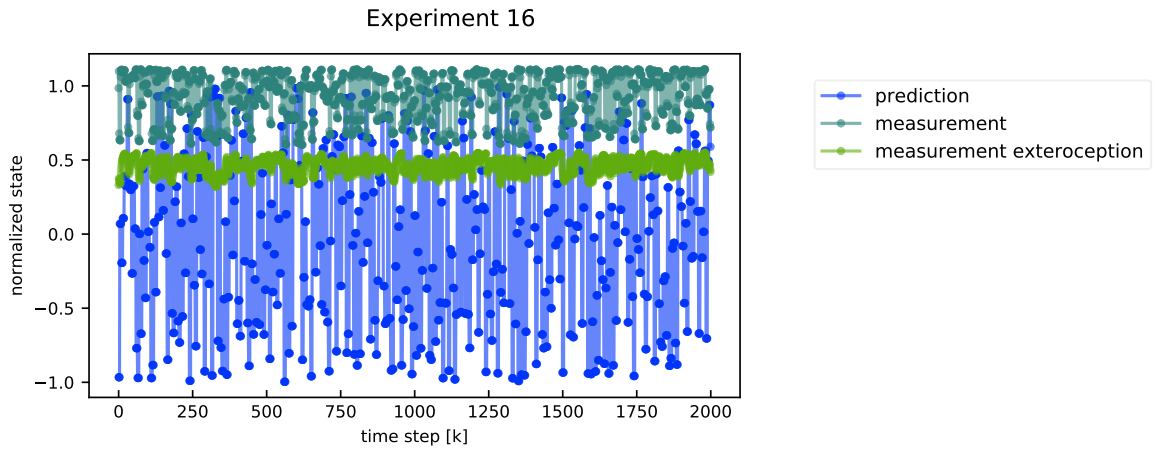



Figure 7.19.: Experiment 16-1 Timeseries of the motor values (proprioceptive prediction) $\hat{s}_0$ in blue, the proprioceptive sensor measurement $\check{s}_0$ in dark green, and the first order exteroceptive state variable (velocity) in bright green. The distribution of the variables is seen as three color bands in the plot. The dissipative term of the velocity (friction) keeps the velocity within bounds while it is still dominated by the remaining inertia. The dissipation parameter is set to 0.2.

sensorimotor data containing force and angular velocity modalities. The data is sampled from a real Puppy robot. Scanning allows to answer questions about information dependencies in spatially extended interaction networks, for example transmission delays from an agent's actions to its sensors. The delays are mostly caused by embodiment, and to a lesser extent, by the larger containing environment. The hypothesis is, that *selectively choosing the model inputs with respect to time and modality by information coupling criteria reduces training time and increases performance*. To test it, the dataset is scanned and the result is used to configure a linear regression probe (Alain and Bengio 2016) for each scan type. A fixed size contiguous window is used a baseline comparison.

### Experiment 17: Puppy periodic slow

A real world robot example is the Puppy robot, initially proposed in (Iida and Pfeifer 2004). There exist several proposals for modifications of the original design. Here, a soft legged design by Andreas Gerken is used, which is described in more detail in (Gerken, Berthold, and V. Hafner 2017). The initial question was, what is the motor-sensor delay of this robot, measured in units of sensorimotor loop steps. The answer is that at the given loop frequency there is no single global delay but rather a set of delays spread out in time. This is caused by differences in speed of information propagation through the robot body. In particular, propagation speed is frequency dependent.

The experiment consists of a data source and a maximum window size prior. Three scans are performed with three types of multivariate *global* measures that differ in how they account for multiple channels of coupling. Global means that all source- and destination variables are each

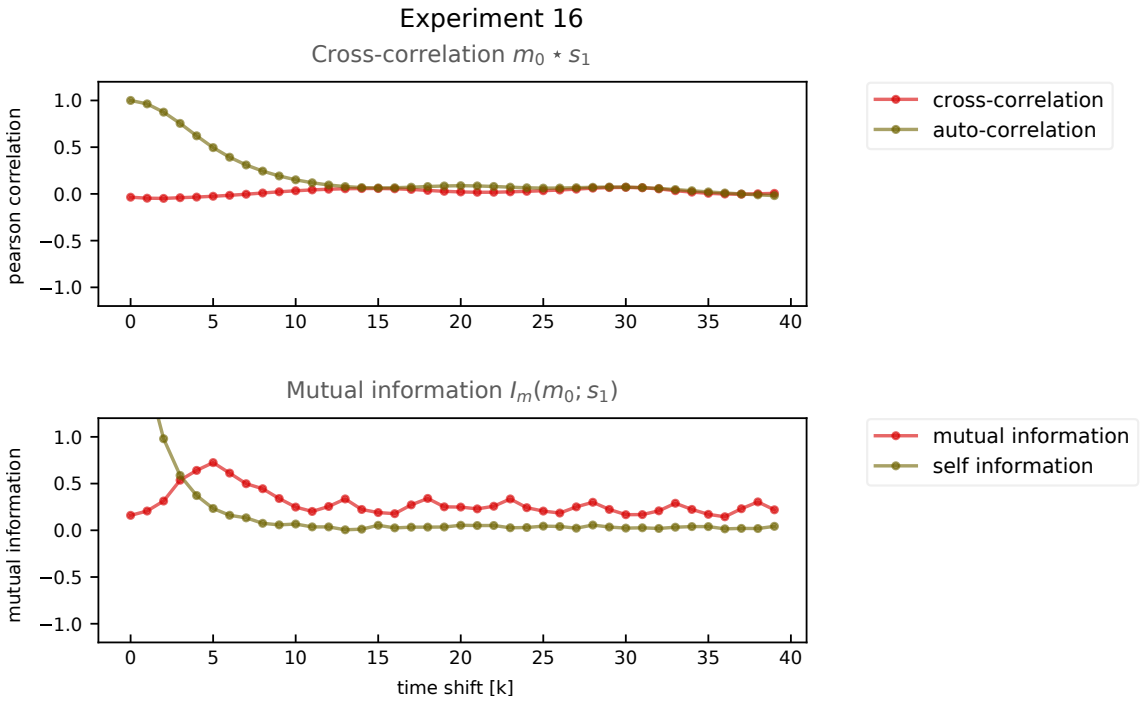



Figure 7.20.: Experiment 16-2 Results of a correlation scan (top) and an information scan (bottom). Normalized correlation coefficients take on values in the interval $[-1, 1]$. The mutual information is unnormalized in the range of $[0, 1.6]$. The mutual information captures the interaction between action and velocity which is not the case for cross-correlation. The auto-correlation and the self information in the plots quantify the amount information the variable has about itself over time. Due to the nature of the system this is at a maximum at zero shift and monotonically decreases with increasing shift. The self information curve is clipped to maintain a scale where the mutual information is well visible.

lumped together to compute the shared information for each time shift. The scan result is a vector scan with each scalar element $\text{scan}_i$ being a dependency measurement for the corresponding time shift of $-i$ of the destination with respect to the source. The learned tappings are compared with a rectangular window baseline using linear regression probes (Alain and Bengio 2016). If the effective coupling is sparse within the window, the tapped input outperforms the baseline probe measured via the mean squared prediction error. In addition, the sparsely tapped probes have significantly lower parameter norms when the regularization parameter is set to a low value, e.g. here $\alpha = 0.01$. The regression probe results are shown in Table 7.1

In all runs up to Experiment 22, the same signal is sent to all four motors of Puppy. Here, the mtotor signal is a square wave with an amplitude range of $[-0.2, 0.2]$ and a period of 76 time steps. The scan length is set to twice the period length. The periodicity is clearly visible in the mutual information measurement, which is causally spurious but statistically correct. This is due to the periodicity of the source. The raw sensorimotor timeseries is shown in Figure 7.21 together with a the histograms to help characterize the episode. The top row contains the onboard acceleration and rotation measurements, the motor signal is shown in the bottom row. The scan results of three different information measures are shown in Figure 7.22. The measures used are the mutual information and conditional mutual information using two different conditions. In the center plot labelled CMI, the condition is the motor past. In the right most panel the transfer entropy is used, which conditions the output on the past of the destination.

For all three scans the results are drawn as points over time shift. In addition, a quantitative tapping is illustrated. Tapped points are shown in red and those ignored are drawn grey. The tapping selection is computed by sorting the scan results, computing the cumulative sum up to a threshold $p$ and then selecting the indices in the returned sum up to the threshold. Here, the theshold is set uniformely to $p = 0.3$ for all experiments, and generally $p \in [0, 1]$ for normalized information.

Table 7.1.: Experiment 17 results table for linear regression probes on four different configurations. The tapping specifies which indices from the input window are used as inputs for the linear model. The baseline of a contiguous window over the entire scan length is compared to the three quantitative tappings derived from mutual information measures. The columns are configuration, residual error (RMSE), regression weight norm ($|W|$), and regression bias norm ($|b|$). The residual error is approximately equal for all four conditions. For the baseline this comes at the cost of a slightly increased weight norm. In this episode, the system is mostly at rest and little information is transferred in total. The effect will be more pronounced in the following experiments.

| **Tapping** | **RMSE** | $\lvert\mathbf{W}\rvert$ | $\lvert\mathbf{b}\rvert$ |
|---|---|---|---|
| Baseline | $0.107$ | $2.19$ | $2.19$ |
| Mutual information | $0.105$ | $1.85$ | $1.85$ |
| Conditional mutual information | $0.107$ | $1.02$ | $1.02$ |
| Transfer entropy | $0.106$ | $1.35$ | $1.35$ |

### Experiment 18: Puppy periodic fast

The otherwise unmodified Experiment 17 is repeated on a different dataset, recorded using a faster motor oscillation with a period of 26 time steps. The results are shown and discussed in Figure 7.23, and Figure 7.24.

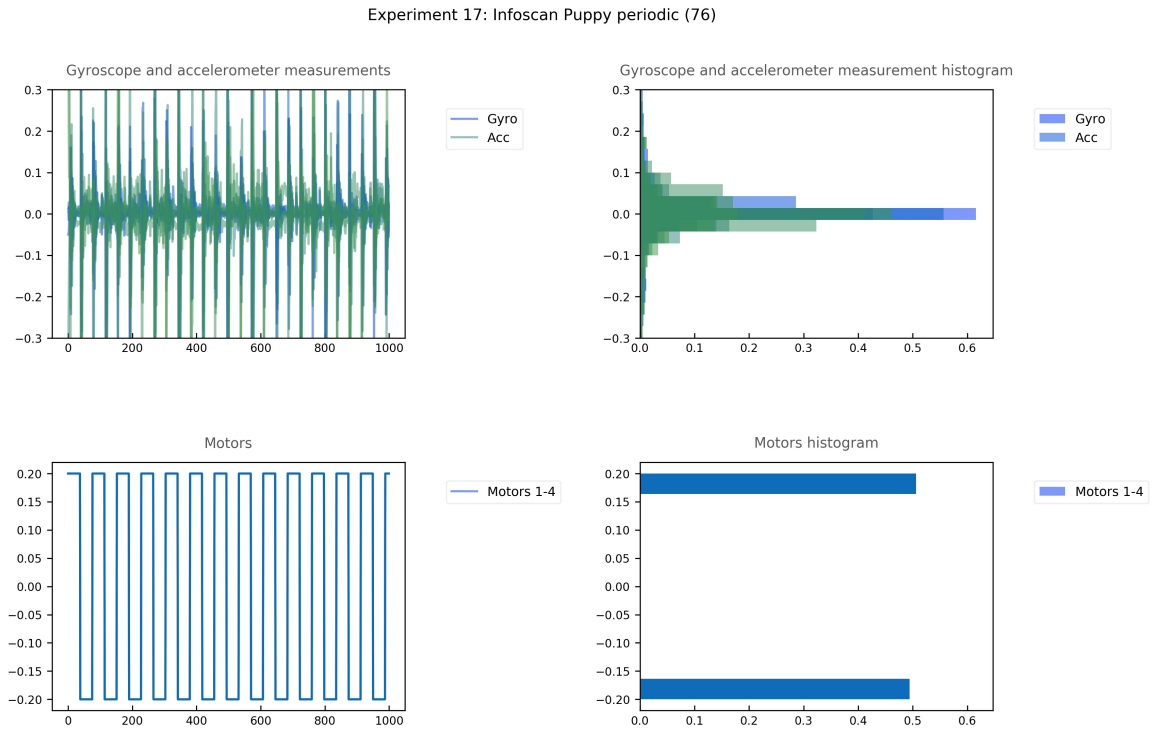



Figure 7.21.: Raw sensorimotor data of Experiment 17. The top row contains the gyroscope and accelerometer sensors in blue and pale green, and the motor signal is shown in bottom row. For both rows, the timeseries is in the left column, and the histogram to the right. The motor signal is sharply distributed between two discrete values. The period is is just long enough to let the system return to resting state.

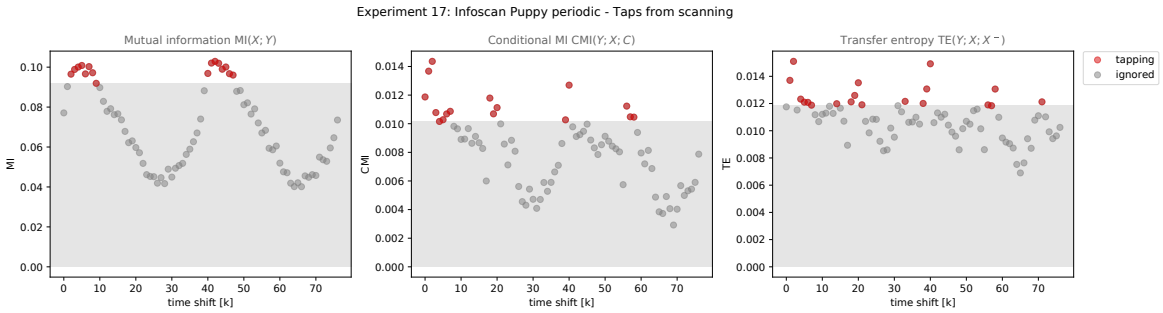



Figure 7.22.: Three information scans over the data shown in the plot above. The scan result is shown as a series of points. A quantitative tapping is computed via the threshold method described in the text. The white horizontal band at the top and red points indicate the range of shift values contributing the most important 30% of information transferred from motors to sensors within the scanning interval.

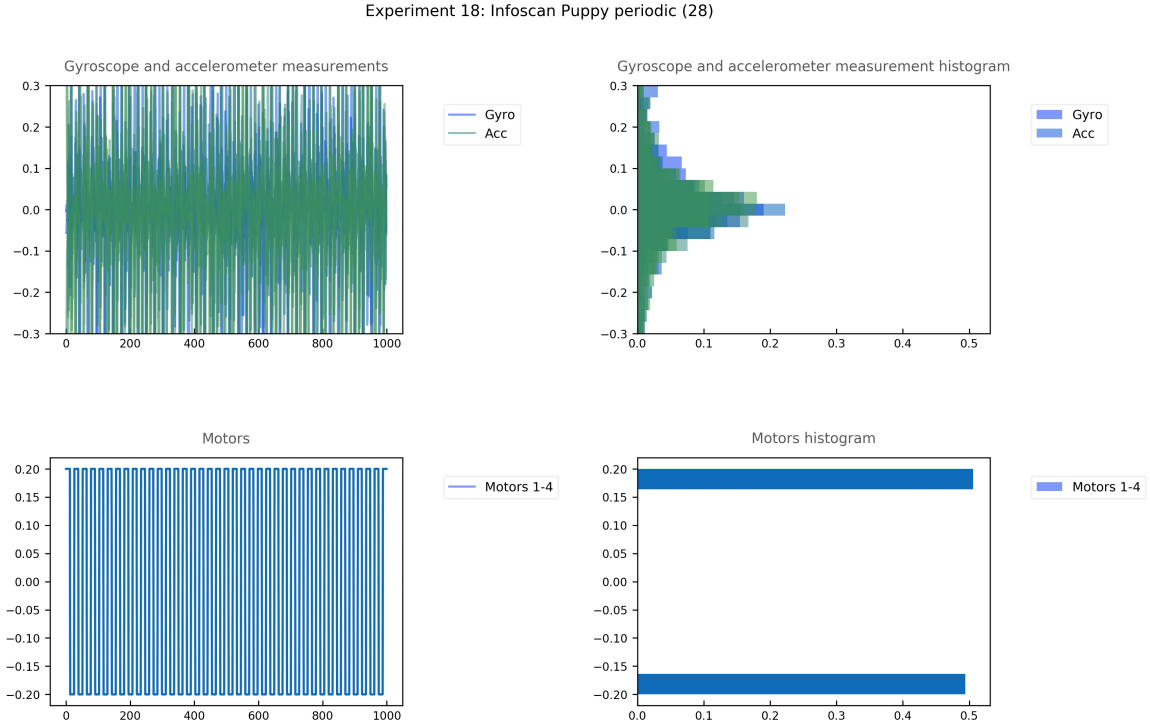



Figure 7.23.: The sensorimotor data for Experiment 18 with sensors in top row, and motors in the bottom row, timeseries left column, and histograms in the right column. It can be seen that motor signal oscillates faster which leads to a larger spread of the sensor values and potentially more information to be transferred.

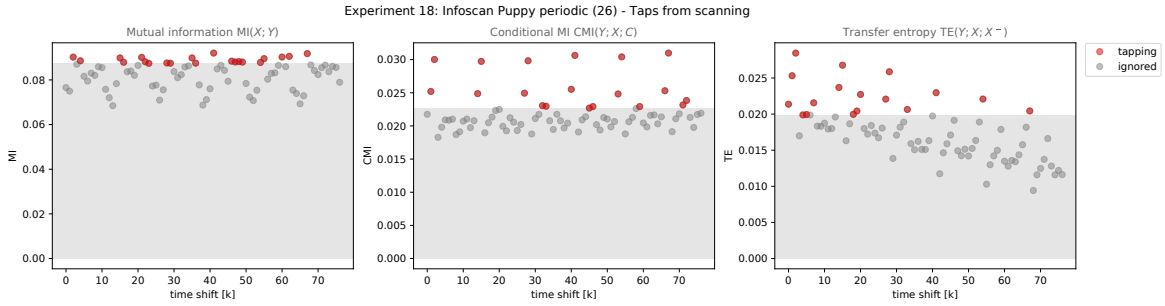



Figure 7.24.: Computed tappings for Experiment 18. The plot is familiar in principle from the preceding experiments, the effective tapping computed for each measure is highlighed by the red points. The shift values that are ignored are covered by the grey band.

Table 7.2.: Regression probe results for Experiment 18 for a contiguous window baseline and three quantitative tappings derived from mutual information measures. The columns are configuration, residual error (RMSE), regression weight norm ($|W|$), and regression bias norm ($|b|$). The baseline residual error is considerably larger than the in the tapped cases. In particular the parameter norms for the baseline are an order of magnitude above those of the tappings. Smaller weights in general will lead to more benign predictions.

| Tapping | RMSE | $\|\mathbf{W}\|$ | $\|\mathbf{b}\|$ |
|---|---|---|---|
| Baseline | 0.18 | **10.35** | **10.35** |
| Mutual information | 0.16 | 1.90 | 1.90 |
| Conditional mutual information | 0.16 | 1.82 | 1.82 |
| Transfer entropy | 0.17 | 2.31 | 2.31 |

#### Experiment 19 Puppy motor frequency sweep

Again Experiment 17 is rerun on a different dataset. This time the data is taken from an episode of a smooth sinusoid motor signal being frequency swept from 0 to 6.4 Hz. The full episode is 5000 time steps in length, here only the first 1000 time steps are considered. The sinusoidal signal generates a broader distribution of values and due to the smoothness of the signal, the predictability should increase. The results are shown according to the familiar pattern of the previous experiments in Figure 7.25, Figure 7.26, and Table 7.3.

Table 7.3.: Regression probe results for Experiment 19 for a contiguous window baseline and three quantitative tappings derived from mutual information measures. The columns are configuration, residual error (RMSE), regression weight norm ($|W|$), and regression bias norm ($|b|$). The residual error is low in general compared to the square wave condition but the baseline error is twice as large as in the tapped cases. Again the parameter norms for the baseline are an order of magnitude above tappings. Tapping could thus also be interpreted as a regularization.

| Tapping | RMSE | $\|\mathbf{W}\|$ | $\|\mathbf{b}\|$ |
|---|---|---|---|
| Baseline | **0.10** | **9.70** | **9.70** |
| Mutual information | 0.05 | 1.24 | 1.24 |
| Conditional mutual information | 0.05 | 0.91 | 0.91 |
| Transfer entropy | 0.05 | 0.91 | 0.91 |

### 7.3.6. Infoscans

Using a sliding window to measure statistical dependency with any information measure will be called an **infoscan** in the remaining section. In the the same scope, specific measures will be supplied in context and are either mutual information or some particular conditional mutual information.

All the results of the preceding experiments are *global*, both in terms of the location within the episode that they occur and in terms of which individual variables contribute which amount to the outcome. The sweep episode of Experiment 19 only uses the first 1000 time steps of a longer sweep episode. From first principles it can be expected that the coupling delays in a robot like Puppy are frequency dependent, and to perform the measurement, the existing setup can

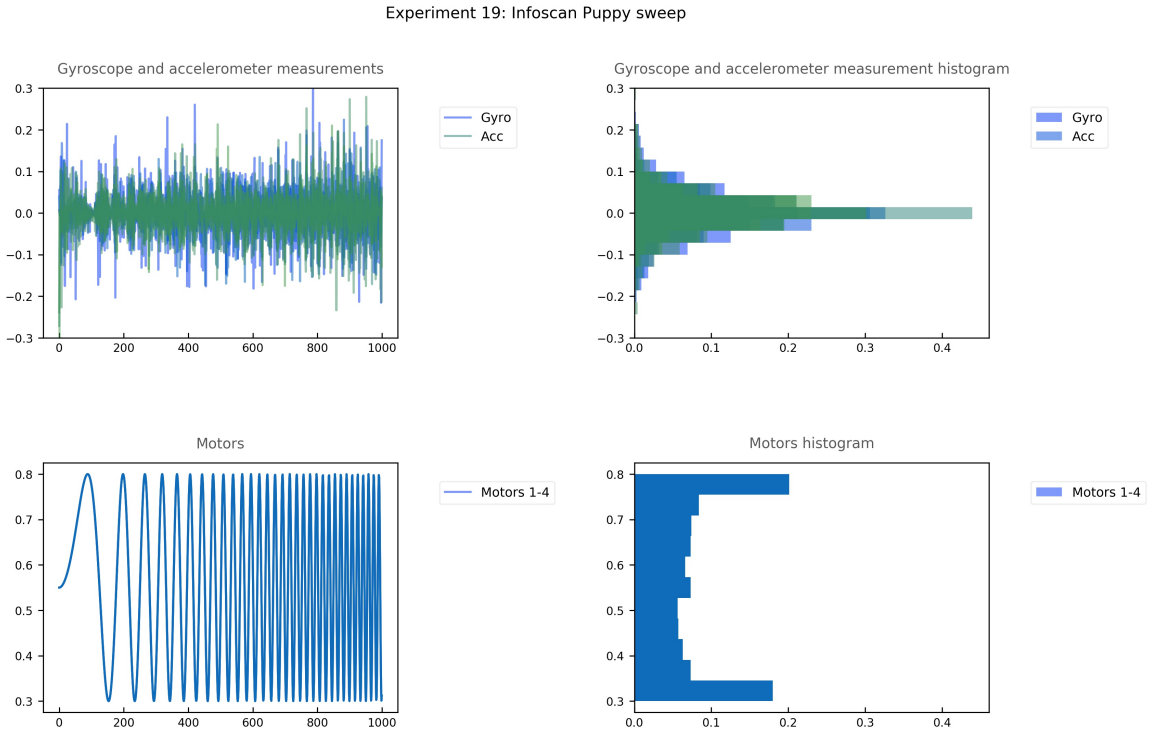



Figure 7.25.: Experiment 19-1 raw sensorimotor data with sensors in shown the top row, and motors shown in the bottom row, timeseries left column, histograms right column. The motor signal sweep is clearly visible in the bottom left of the figure, leading to a broad distribution of values in the histogram to the right. The sensory response shown in the top left panel has lower peak amplitudes and spread as the square wave condition.

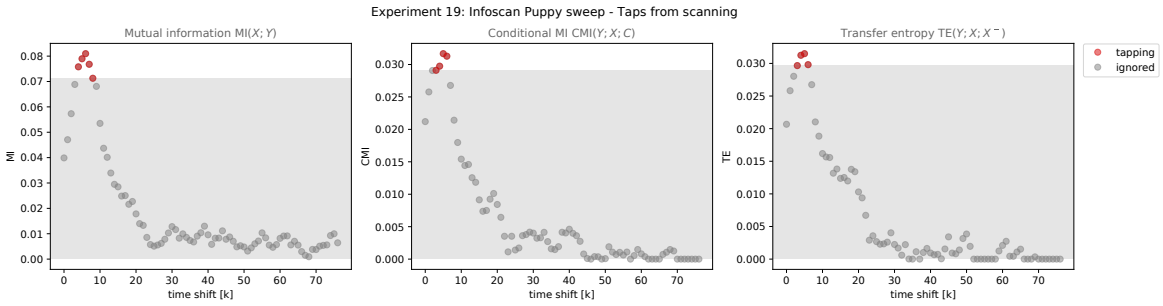



Figure 7.26.: The effective tapping computed for Experiment 19 for each measure is highlighted by the red points. The shift values that are ignored are covered by the grey band in the lower part of each plot. The response peak for the sweep signal is much more pronounced than in the periodic condition leading to even lower parameter norms for regression probes.

be extended in a straightforward manner with the sliding window method. This method is used pervasively and has many alternative names depending on the domain and context, for example delay embedding, short-time transform, fractional differentiation, etc. Experiment 19 is repeated with two modifications. This time, the entire dataset is used with a total length of 5000 time steps. This increases the range of motor frequencies over which the response is observed. The original size of the analysis window of 1000 time steps is reduced to 500 steps and the resulting window, together with the attached measurements, is repeatedly applied at consecutive window-size integer multiples up to the new episode length. This results in episode-length/windowsize different measurements, each of length $\int^{|\mathsf{win}|} tdt$. The measurements are localized because they only use a finite amout of information (the window) in the vicinity of the current windowing index. The final three experiments in this chapter serve to illustrate infoscans.

**Experiment 20: Windowed infoscan Puppy sweep**

This experiment consists of the full open-loop frequency sweep Puppy exploration dataset of 5000 time steps. Information scans are applied repeatedly on a window sliding over the dataset with a step size equal to the window size (500 time steps). The experiment is intended to highlight the fact that the information propagation delays in moderately complex robot bodies, like Puppy, can be time dependent. In the experiment, the motor frequency sweep ties together time and frequency and each window's measurement is in direct correspondence with the frequency range swept within its window. The timeseries and histogram plot is omitted and is similar to Experiment 19.
The scan result shown in Figure 7.27 is two-dimensional and rendered as a heatmap. The time index is on the vertical y-axis with the shift remaining on the x-axis. Each row represent a scanning frame of all shifts for each time in the episode. Dependency measurements are color coded with zero being white and the maximum in dark red. By the progression of the sweep through time each row is implicitly tied to a frequency range. The periodicity of the motor signal leads to a large amount of shared information through periodic overlap, which the mutual information cannot discern from a causal effect. Conditioning the MI on the motor and sensor past allows to remove this effect. This is visible in the middle and right hand panel where the second and third row clearly indicate a maximum of information transferred in comparison with other motor frequencies. This can be interpreted by the agent as an approximate resonant mode of the robot body and provide a starting point for further self-exploration that is likely to quickly yield a large degree of control.

### 7.3.7. Element-wise scanning

Another axis available for scanning is modality, that is, the individual motor and sensor channels. Modality in scans has been introduced in the periodic- and sweep motor signal Experiments 21 and 22. The exploration strategy is the same as in previous experiments but this time the dependency of sensor variables on the motor variable(s) is measured individually for each motor-sensor pair. The results are in agreement with the expected differentiation of coupling strength among the different modalities. In terms of the self-exploring agent this an even sparser hypothesis for the

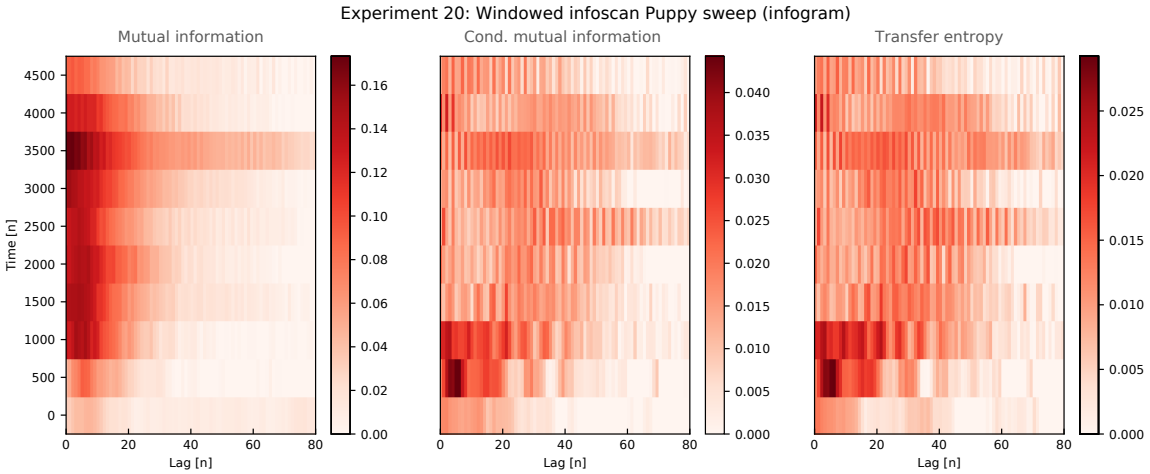



Figure 7.27.: Experiment 20 is a sliding window infoscan for over the full-length sweep dataset. The measures used in the scan from left to right are mutual information (MI), conditional transfer entropy (CTE), and transfer entropy (TE). The condition for the CTE is the source (motor) past, for the TE it is the destination (sensor) past. The mutual information cannot distinguish between apparent and causal interactions and measures a large amount of shared information that is in fact caused by periodicity. Both the CMI and the TE improve the measurement significantly with respect to finding better candidates of true causal interaction.

currently effective motor-sensor coupling than the global measure. The computational overhead for this additional degree of detail needs to be in balance of course with the benefits in model size and learning speed.

### Experiment 21: Windowed element-wise infoscan Puppy periodic

A windowed and element-wise infoscan is performed over time shifts as before and over all motor/sensor variable pairs. In general the result is a tensor of shape $(d_{\text{motor}}, d_{\text{sensor}}, d_{\text{lag}})$ which needs to be embedded for visualization in a matrix with lag columns and motor $\times$ sensor rows. Here the situation is simpler because all four motor signals are identical, so only one panel is needed. The MI in the leftmost panel is dominated by the longitudinal acceleration sensor (acc_y). This is a spurious measurement with respect to the actual coupling which is seen in the measurements conditioned on the destination and source variable's own pasts in the middle and right hand image in Figure 7.28.

### Experiment 22: Windowed element-wise infoscan Puppy sweep

Experiment 22 repeats the elementwise scans of Experiment 21 with the sweep exploration dataset. The scan result is shown in Figure 7.29. The scan highlights additional details about the motor-sensor coupling of the Puppy robot. The low-frequency resonances are not produced under this condition, making the mutual information agree more with the conditional measures. The sweep signal's actual transfer of information is larger than for the high bandwidth square pulses of

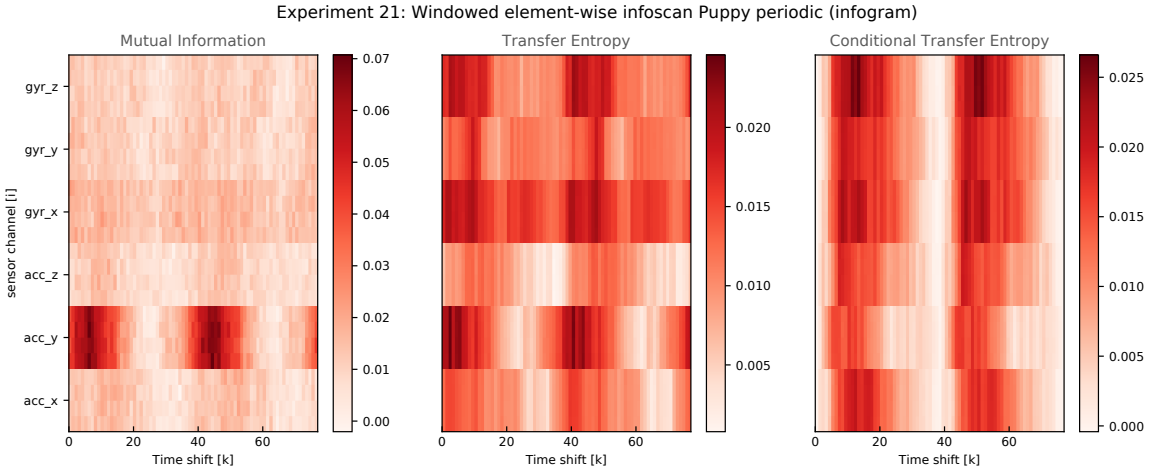



Figure 7.28.: Experiment 21-2 Pairwise infoscans for each of three dependency measures MI, TE, and CTE. The large MI in the leftmost plot is caused by body or sensor resonances from the low-frequency component of the motor signal and not by the momentary action. This is accounted for by the conditional measurement variants of TE (conditioned on the destination's past) and the CTE (additionally conditioned on the remaining three motor signals). It can be seen that information is transferred most quickly to longitudinal acceleration. This axis corresponds with a rotation around the transverse body axis, showing up in the x-axis of the gyroscope. The delay in comparison with the acceleration is to be expected from the order relationship of the variables. Most interesting is the yaw rotation (gyroscope z-axis) which should not occur ideally and results exclusively from asymmetries in the embodiment.

the periodic exploration signal, as can be read of the respective color bars. Also, information is transferred as a compact packet instead of the intermittent response to the square pulses. This demonstrates that an agent's action-delay expectation is a dynamic entity with a potentially important role for both introspective (self-state) as well as predictive functions.

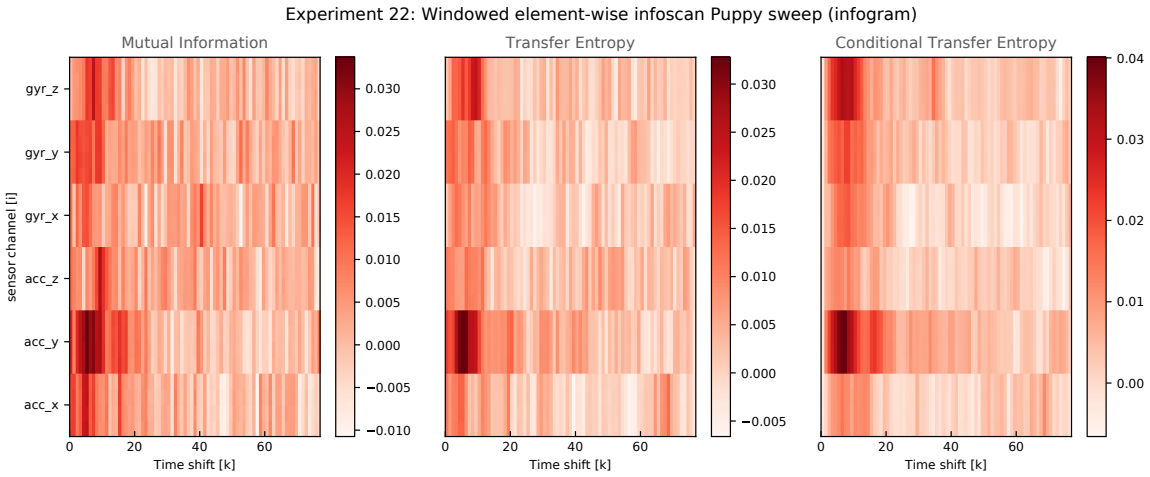



Figure 7.29.: Experiment 22-2 Pairwise infoscan for each of three dependency measures, the MI, TE, and CTE. In comparison with the previous analysis of Experiment 21 the sweep exploration seems to elicit a clearer result. All three infoscans are in qualitative agreement. The acceleration along the longitudinal body axis is affected most by the motors, which is to be expected from the design. The information propagates through the system and shows up in the gyroscope measurements. Again, there is a large amout of effect on the yaw rotation which results from small physical asymmetries. Conditioning out additional motors in the CTE configuration makes the yaw interaction specific for a particular motor.

### 7.3.8. Discussion

Infoscans are a method to learn body maps on any kind of robot embodiment. They allow an agent to answer questions internally that otherwise need to provided as design priors from the outside. In their conditional forms, infoscans allow to make educated guesses about the precise timing and task-relevant modalities for embodied robots. Thresholding the scan output provides an effective algorithm for learning a tapping in situ that can be used as a prior in subsequent skill learning. The tapping prior provides a model agnostic regularization during later learning learning stages, which can improve the learning speed, prediction performance, and generalization.
Considering different measures proposed in the literature relation, each appears as a particular conditioning of mutual information. Starting from a global measure, the primary scanning dimensions of time and modality, allow to generate measurements localized to within a given window size. The behaviour of these measures under change of the integration parameters themselves (the window size) was identified as a promising *aggregate* measure of the self-generated *complexity*, and is discussed in detail in (William Bialek, Nemenman, and Naftali Tishby 2001). The tapping

hypothesis is, that learning is facilitated when the structure of input dependency is known before training an internal model. Infoscans allow to learn a tapping for any sensorimotor context, in case it cannot be provided a priori.
The main operators for building aggregate measures are *integration* and *differentiation*. In their textbook versions, the integration operator is completely global and the differentiation operator entirely local. Fractional calculus provides tools to configure the locality of these operators. Localized operators can be translated into finite impulse response filters for the discrete case with the corner cases of ideal low pass and high pass filters for exponents 1 and -1 of the differential-integral operator. Fractional exponent values result in bandpass filters. These operators allow to construct precisely those measures that are needed for autonomously learning agents to modulate the motivation state.
Since the TE was originally proposed as $TE(X \to Y) = m_I(Y_t + 1; X_t|Y_t)$ additional improvements and variations have been proposed, for example the transfer entropy with interaction delay $TE(X \to Y, u) = m_I(Y_t + u; X_t|Y_t)$, the momentary information transfer $MIT(X; Y, u) = m_I(Y_{t+u}; X_t DY_{t+u-1}, X_t - 1)$, or the self-prediction optimal transfer entropy $TE_{SPO}(X; Y, u) = m_I(Y_t; X_t - u|Y_t - 1)$, to more precisely disentangle the directions, delays and causality of interactions (Wibral, Pampu, et al. 2013). The MIT has been used to quantify the propagation of information in a soft robotic arm (Nakajima, Schmidt, and Pfeifer 2014). A comprehensive framework of information dynamics, that is, the flow, storage, and processing of information in dependence of where and when it occurs, is presented in (Lizier, Prokopenko, and Zomaya 2014). The corresponding toolkit (Lizier 2014) is used to perform the infoscans in the current work.
Information theoretic analysis of *multivariate* timeseries is not completely understood, despite such significant progress. A recent proposal is partial information decomposition (PID) (Williams and Beer 2010), which specifically addresses problems in earlier ones by identifying and disentangling mutlivariate interactions into *unique*, *redundant*, and *synergistic* components. The PID method has been applied to quantify distributed computation in neuroscience and robotics, for example to express goal functions of neural computation independently of their domain (Wibral, Priesemann, et al. 2015), or to quantify the *morphological computation* that occurs within the sensorimotor loop of an embodied agent (Ghazi-Zahedi and Rauh 2015).
The *generalization of several special cases of measures*, which are all based on the conditional mutual information, and which have an intrinsic notion of time, *emerges* as a *byproduct* of the infoscan algorithm. The idea is not explored further beyond some of the immediate needs of learning a tapping for an sensorimotor models.

### 7.3.9. Conclusion

In this chapter the tappings prior of section 7.2 was complemented by a method for learning an actual tapping from data based on information theoretic measures. Several such measures and a principled decomposition of multivariate information have been proposed in previous works. The method described here is a generalization of conditional mutual information measures where explicit choices of variables, delays, and embedding configurations in existing variants are treated as special cases of continuous parameter ranges which can be integrated over to provide additional information about sensorimotor dependencies not available from any one single configuration

point. In addition, the presence of a sensorimotor loop prior in the artifical life and robotic context entails valuable constraints that simplify the information decomposition.

## 7.4. Results

The two major contributions in this chapter are *tappings* and *infoscans*. Tappings are an abstract probabilistic graphical concept for embedding adaptive models in a real time sensorimotor data stream. Infoscans provide an empirical counterpart to tappings. It has been shown that an infoscan can be turned into a learning algorithm for tappings by using a threshold on the sorted and summed dependency measurements. The next and final chapter of this part rests on the assumption that such a mechanism is in place on top of which the skill learning can take place quickly and efficiently.

# 8. Skill acquisition

In this chapter three different variations of a general developmental models are presented. All of them are real-time closed-loop skill acquisition models that operate on raw sensorimotor data streams. They are presented experimentally and analyzed qualitatively,.

## 8.1. Developmental models

The developmental approach to synthetic agents is motivated by the realization, that existing and classical approaches to building and designing agents and robots are insufficient. In particular, this is the case when regarding the challenges of building robots that are complex in themselves and that face complex environments populated with natural agents. A counterintuitive consequence of these complexity challenges is that it becomes more important for agents to fail smoothely instead of catastrophically over many different tasks, than to perform optimally on any single or isolated tasks.
Within the scope of learning algorithms, the developmental approach is different in category from the well-defined domains of machine learning like supervised and unsupervised learning. The main distinction is that the developmental problem needs to tackle the question of actively generating the training data on the go, while low-level learning is still in progress. This is the exploration problem, as it is seen from the learning point of view. The field of reinforcement learning is a well-known branch of the developmental approach, although RL itself it is not necessarily accessed from the developmental point of view. A developmental model thus becomes at the core a *composite, feedback controlled process*, that takes an agent through an appropriate sequence of smaller learning problems, that work together to provide a basis for a robust and highly adaptive overall strategy, able to cope with multiple objectives, and difficult exploration and representation learning issues.
The current approach rests on the hypothesis, that robust, large scale intelligent behaviour needs to be implemented in a massively distributed way, where each of the constituent components are mildly intelligent in their own, very limited micro-environment. These are known for example as mixtures of experts, where a gaussian mixture model can be seen as an example mixture of local gaussian experts. The focus rests on one such expert unit and how it can control its own adaptation via local exploration, local learning and and integration of locally impinging feedback from the surrounding context of other experts and support structure.
Three main families of mechanisms are considered in this chapter. The first takes the idea of forward/inverse internal model pairs as a basis. It is shown how such a pair can be utilized by an agent for very coarse batch-based learning and control which is close to well-known machine learning methods. The main contribution in this subsection is a family of algorithms to perform online direct forward-inverse model learning. The core algorithm is described in detail and the

main characteristics are shown in toy model experiments. More complex robotic use cases and algorithmic variations can be found in the Robot experiments chapter of the appendix.

## 8.2. Internal model online learning (imol)

The classic scheme is a pair consisting of a forward and an inverse model and is shown in Figure 8.4. The forward model is forward with respect to sensorimotor time and causality and acts as a predictor in the classical sense of predicting the future considering the known state and action history, in particular the privileged knowledge of the action just committed. The inverse model inverts time and causality by predicting a cause leading to an effect where the cause is the agent's own action and the effect is a desired state, which is given by some top-down prediction, usually called a goal sampled from motivation. In classical language, the inverse model provides active, goal directed access to state space. In mechanical terms it is simply a predictive motor code.
This is straightforward on a conceptual level but there are some intricacies to consider in the detailed mechanics. Issues of external context have been discussed at length in the preceding chapter on self-exploration. Internal issues of the model's representational power concern its own uncertainty. Uncertainty contains ambiguity (local handling vs. bottom-up delegation of disambiguation), learning rates, and introspection as subproblems. In particular, even if the models are assumed to be pretrained somehow, just the model pair itself is not sufficient for obtaining a fully operational module. The actual input assignments have to be given along with some additional circuit controlling the actual inference at a performance level exhausting the locally available information. In the bottom diagram in Figure 8.10 an example circuit is drawn in red. That circuit implies a local iterative search process, that tries to consolidate the predictions. In abstract terms, the synergistic information available in both submodels taken together is integrated to obtain a more robust overall prediction.
In the batch training setup, the agent is either supplied with existing sensorimotor data, or uses an exploration strategy that is fixed over the exploration episode. At the end of the episode, the internal models are fitted to their respective training data. In the second pass of the agent's lifetime, it exploits the acquired models, assuming that learning was successful in terms of the model adequacy to agent tasks. This scenario is described in more detail C.1 chapter of the appendix and the open-loop learning process is shown in Figure 8.1.
This procedure can be problematic, when the environment is too complicated for the initial exploration strategy to generate any meaningful data, simply because it is too uninformed. If the exploration strategy is made more informed, the original learning goal is obsolete. This is a chicken and egg type problem. In fact, the two passes of 1) fitting data, and 2) exploiting the fitted models, can be run iteratively resulting in a succession of episodes. If the batch size is made increasingly smaller down to the limit of a single time step, the problem turns into that of online learning.
Online learning means to incrementally update a model with a single data point as shown in Figure 8.3. It can also be interpreted as adaptive filtering when regarded from a signal processing or control perspective. The main motivation for online learning comes from the following scenario. An agent might need to bootstrap a minimum amount of control as quickly as possible, to avoid damage due to its own inadequate actions. In this case, the agent wants to make use of every

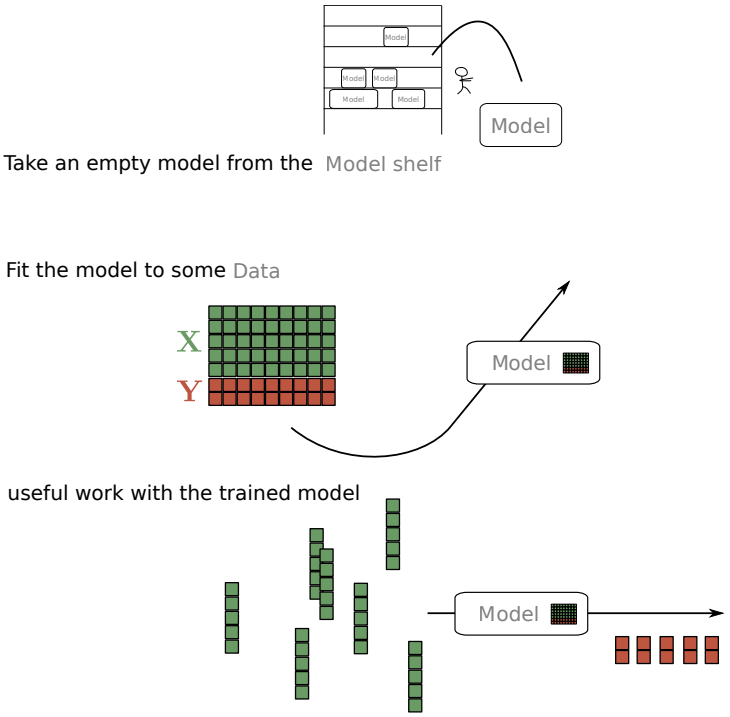



Figure 8.1.: Batch learning.

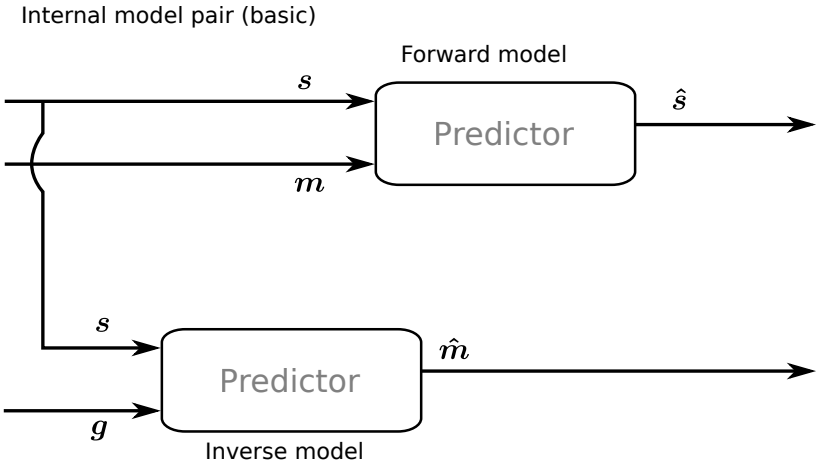



Figure 8.2.: Block diagram of a basic forward-inverse model pair. This basic structure is just a starting point, since the information contained in the diagram is insufficient as a complete working agent specification. In particular, the two models are not interacting at all within the model pair structure itself. Fundamental interaction schemas and their variations will be presented in the rest of the chapter.

bit of incoming information as quickly as possible. This is a clear case where the objective changes from an optimization problem to an adequacy problem, which is a new optimization problem with additional constraints. In every time step, the model is updated with the most recent measurements and the next motor prediction is then based on the new model state. Major questions are, how to represent uncertainty, how to disentangle the confounded uncertainties of the model and of the environment, and how to sample exploration moves from the model in an optimal way.

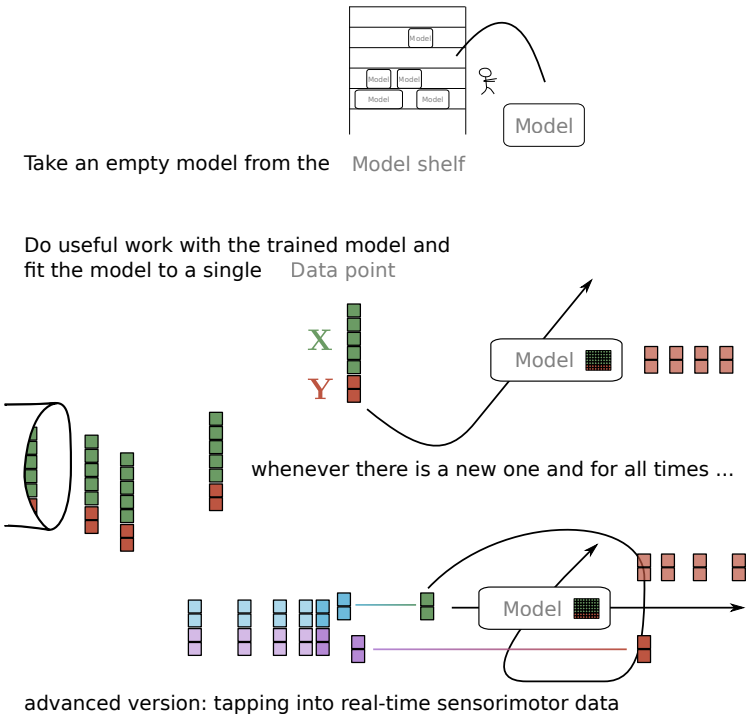



Figure 8.3.: The online learning process.

The base model consists of four main groups of inputs. The goal prediction from the upper level $pre_{l1}$, the state prediction at the current level $pre_{l0}$, the state measurement at the current level $meas_l0$ and the prediction error at the current level $prerr_{l0}$. These signals are routed to two adaptive prediction models, initially of the same kind. Their distinct functional role emerges entirely from the signal routing configuration, the tapping. In particular, the simple block diagram only shows the prediction flow and hides the fact, that the wiring for an update is actually quite different. This is clear from looking at the tapping extracted for the base experiment in Figure 8.5.
Base model parameters and variations, their effects, over selected experiments. Structure and equations. Grounding to primary states, proprioception, and pulling it up. Show hierarchical inclusion as a possibility.
The imol model level of description relies on the fact that the algorithms contained inside the Predictor boxes are fundamentally online algorithms themselves. These type of algorithms effectively establish the connection between adaptive filtering and online learning in the developmental

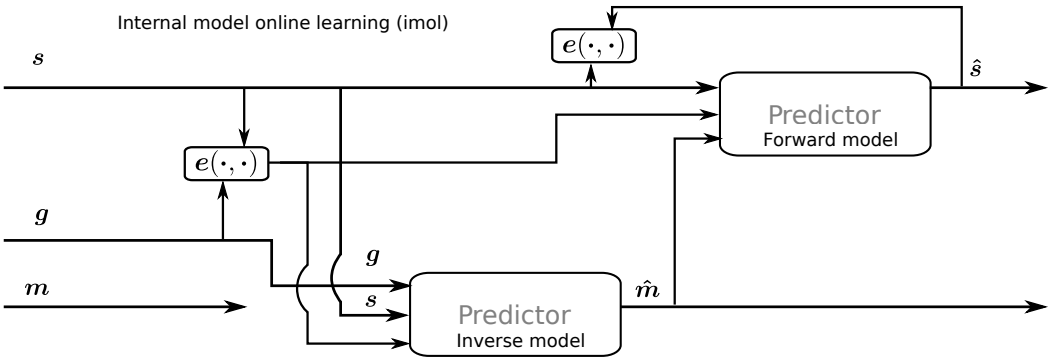



Figure 8.4.: Block diagram of the imol base model.

context. A large variety of such algorithms exist and are used in different variations of the imol model. Examples include the delta rule, all kinds of Hebbian learning rules, the recursive least squares algorithm and its descendants. Low-level learning rules are described in more detail in chapter B of the Appendix.
The algorithm in **Algorithm 1** is in direct correspondence with the underlying smp_graphs code of the dm-imol experiment. This can be verified by tracing the callgraph of the experiment, shown in Figure 8.6, and generated with pycallgraph as shown in Listing 8.1.

Listing 8.1: Call graph generation command with pycallgraph.

```
pycallgraph -v --exclude "*main*" \
  --include "smp_graphs.block_models.ModelBlock2.step*" \
  --include "smp_graphs.funcs_models.*imol*" \
  --include "smp_graphs.funcs_models.model.predict*" \
  --include "smp_graphs.tapping.*"
  --include "smp_base.models_actinf.smpKNN.*" \
   graphviz --output-format dot --output-file dm-imol.dot -- \
   experiment.py --no-ros --no-cache --no-saveplot --no-showplot \
     --conf conf/dm_imol.py
```

Also the tapping can be extracted from the configuration of the experiment. In this case, the precise delay information is available from the configuration of the simulation. In correspondence with the notation of the chapters on graphical and quantitative tappings, the first experiments tapping is shown Figure 8.5. There is one panel for each of the imol submodels, in their actual

**Algorithm 1** The imol algorithm

$\boldsymbol{\theta}_{\text{imol}} =$ ⋯ ▷ Parameters is a configuration tree
$\text{model}_{\text{inv}} = [$ ▷ Populate the configuration tree with live objects
  $\text{update\_prerr\_l0}_{\text{inv}}, \text{tap}_{\text{inv}},$ ▷ update internal state with measurement and tap state
  $\text{smpmodel}_{\text{inv}},$ ▷ fit, predict
  $\text{update\_pre\_l0}_{\text{inv}}]$ ▷ update internal state with new predictions
$\text{model}_{\text{fwd}} = [$
  $\text{update\_prerr\_l0}_{\text{fwd}}, \text{tap}_{\text{fwd}},$ ▷ update internal state with measurement and tap state
  $\text{smpmodel}_{\text{fwd}}]$ ▷ fit, predict
**repeat** ▷ Enter sensorimotor loop, step_imol...
  $\boldsymbol{s} = \text{measure}()$ ▷ obtain measurement and store it
  **for** func in $\text{model}_{\text{inv}} + \text{model}_{\text{fwd}}$ **do** ▷ reduce the composite function stack and call each function
    $\hat{\boldsymbol{m}} = \text{func}(\boldsymbol{s})$ ▷ that the model is comprised of by configuration
  **end for**
**until** end of episode

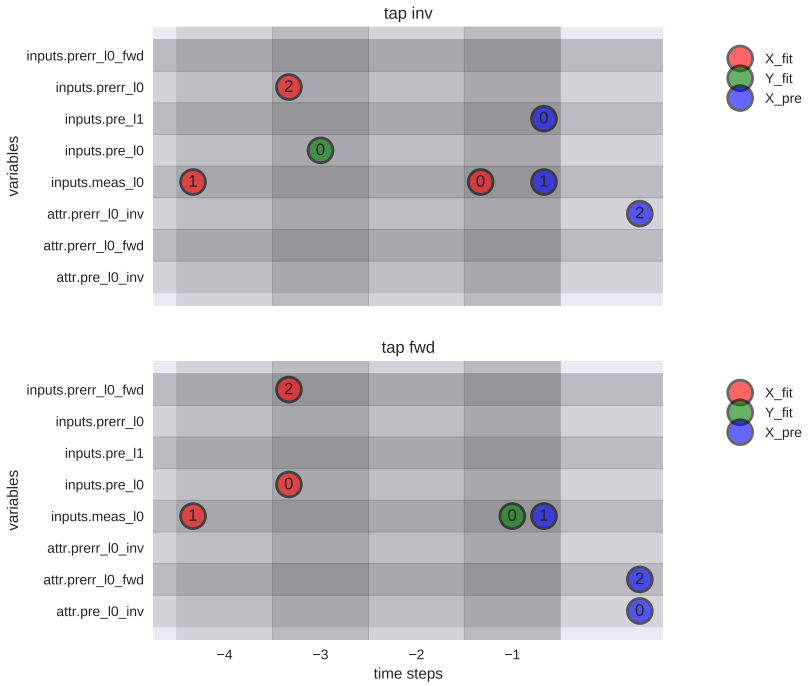



Figure 8.5.: Tapping extracted from the configuration of the dm-imol base experiment.

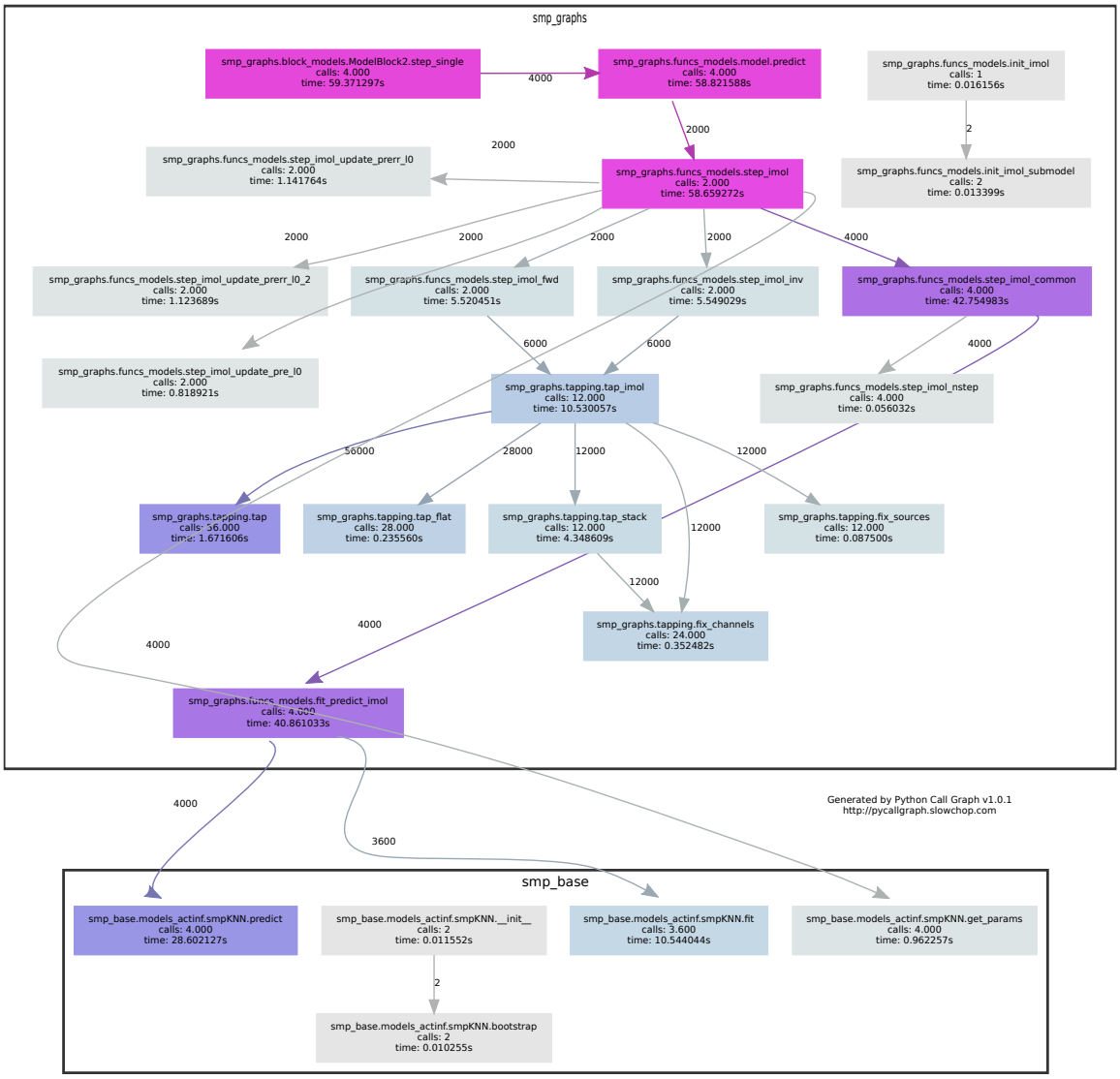



Figure 8.6.: Call graph for the imol model.

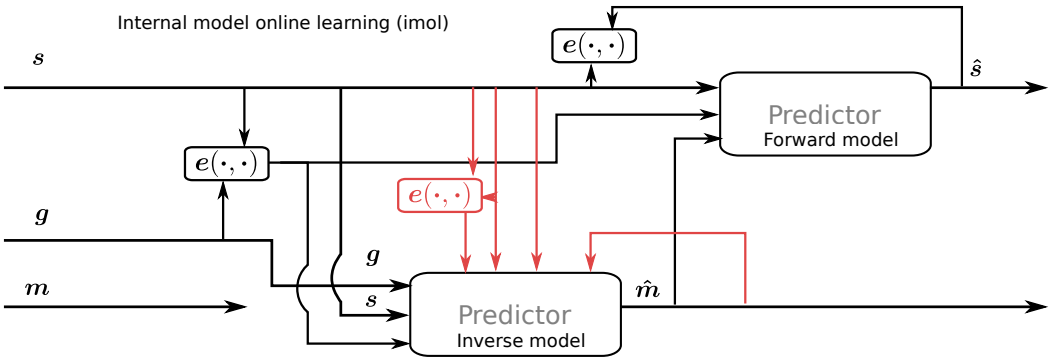



Figure 8.7.: Block diagram of the imol base model including extra lines in red to indicate the signal flow for an inverse model update. This differs signifcantly from the prediction wiring.

order of execution from top to bottom. Each panel contains three groups of nodes, two of which are used for updating the model as input (red nodes X_fit) and target (green nodes, Y_fit), and the other for generating a new prediction with the updated model (blue nodes, X_pre). The row location of a node indicates which variable this represents and the node's number indicates the stacking order of this variable in the raw low-level input tensor of the sensorimotor model.

## Experiment 23 and 24: Internal models online learning

This experiment is a demonstration of the internal model online learning algorithm in a sensorimotor episode. The low-level models in this configuration are using the k-nearest neighbors learning algorithm. The episode lasts for 2500 timesteps. The developmental schedule within that episode consists of bootstrapping the low-level models on initialization (uniform random), a warm-up phase (200 time steps), an actual learning phase (1600 time steps), and a consecutive testing phase (another 200 time steps).
During the learning phase, the following steps are repeated: 1) compute the (inverse) prediction error using the incoming measurement, this prediction error is the difference of actual state and some goal state; 2) the inverse model is fitted to the current error and the corresponding past input, which is still lingering in local memory; 3) predict the next motor command from current goal and state inputs based on the updated model; 4) compute the forward prediction error; 5) fit the forward model with the forward pe; 6) make new forward prediction using current state and

current motor prediction. There is no explicit exploration noise present. The exploration results only from the combined randomness of embodiment and model uncertainty that is present in the system. Here, only the inverse model is used. A timeseries plot of the extended state of the learning agent is shown in Figure 8.8 and discussed below.

## Experiment 24: Internal models online learning variation

Experiment 24 is a variation of Experiment 23 where instead of a discrete goal, a continuous target function is used.

There are six rows of plots shown in both Figure 8.8 and Figure 8.9. The first row is the timeseries of the state error with respect to the top down goal prediction in blue, together with the expected prediction errors for to each submodel in green. The state error is equal to the inverse model's prediction error in this case. The error is large in the bootstrapping phase during the first 200 time steps. This is because the inverse model is only minimally boostrapped from the prior. The forward error is small instead because the system does not move far from the resting state, which is what is predicted on average by the forward model on its prior. The second contains the traces of the goal (top down prediction $\text{pre}_{l_1}$, the state measurement $\text{meas}_{l_0}$, and the forward and inverse predictions. The third and fourth row show the *raw* inverse model input $\boldsymbol{X}$ and target $\boldsymbol{Y}$ after tapping. The tapping used here is that shown in Figure 8.5. The bottom two rows are introspective signals from within the inverse model. In case of the the knn algorithm, the activation is taken as the current dictionary index, and the parameter norm is the total number of dictionary slots used.
The main message of both figures can be read off the first and second row plots. It is the fact that in each case the model acquires perfectly adequate performance within a fraction of the episode immediately after the end of the bootstrapping phase when learning starts. The solid behaviour at the end of the episode that is not visually discernible from the end of the learning phase indicates successful testing, where behaviour is stable without ongoing adaptation. The testing performance better in the continuous goal condition of Experiment 24.
The internal model online learning algorithm has been presented graphically, as an algorithm, and as a fully graphical implementation. Also, it was shown, that the implementation matches the proposed model up to a certain level of detail. Running the experiment in two random configurations showcases the viability of the model for bootstrapping motion control capabilities in an embodied agent.

## 8.3. Active inference

Pairs of internal models have often been formulated and discussed using the established terminology of classical cybernetics and systems theory. The effect of this is seen in the distinction of motor and sensor signals, the use of functional roles like forward and inverse models, and a corresponding loss in generality. This terminology is challenged from within neuroscience by

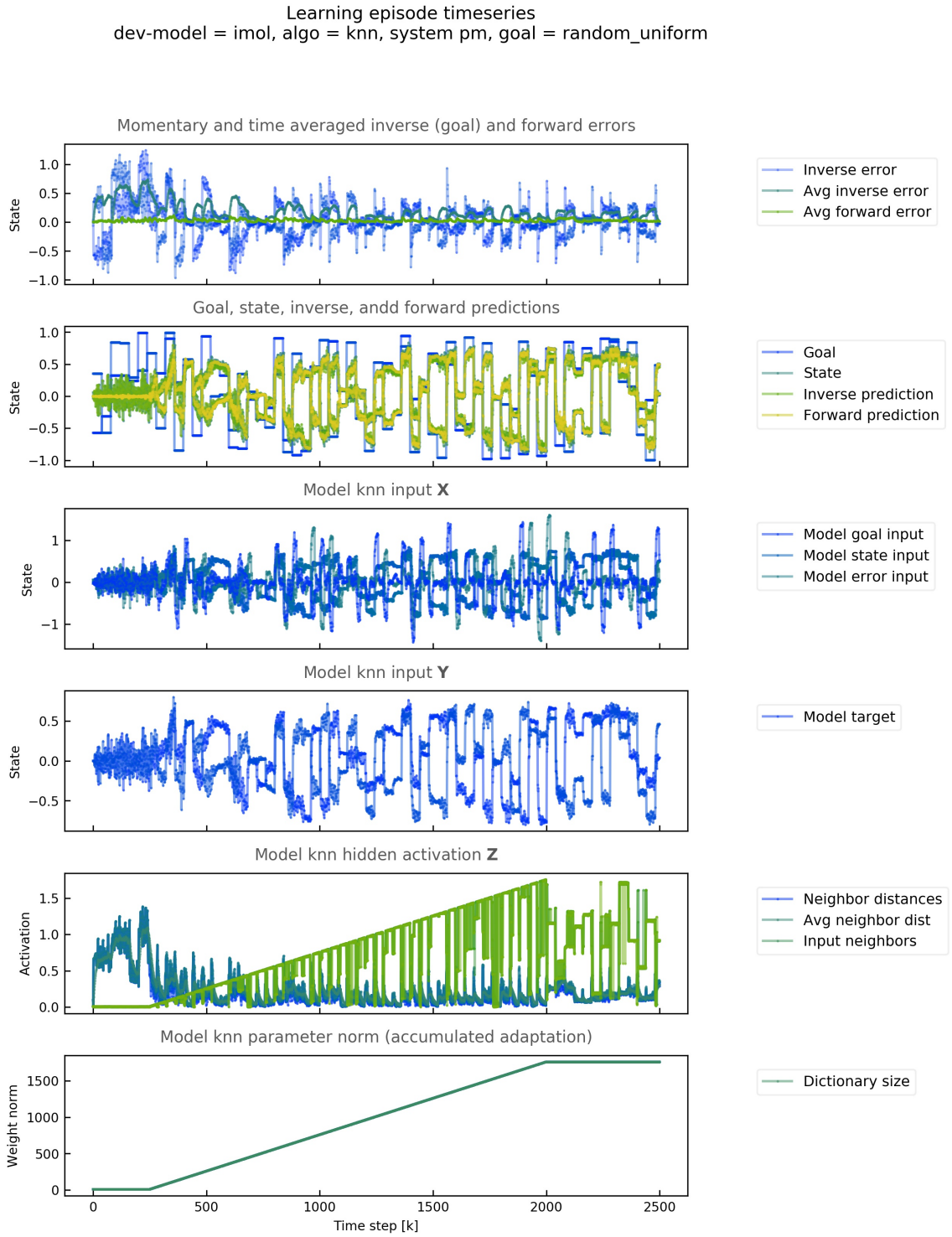



Figure 8.8.: Experiment 23-1: An internal model online learning agent learning to control a two-dimensional point mass system in the discrete goal condition using the knn low-level algorithm. The three phases of bootstrapping, learning, and testing can be read off the bottom two panels.

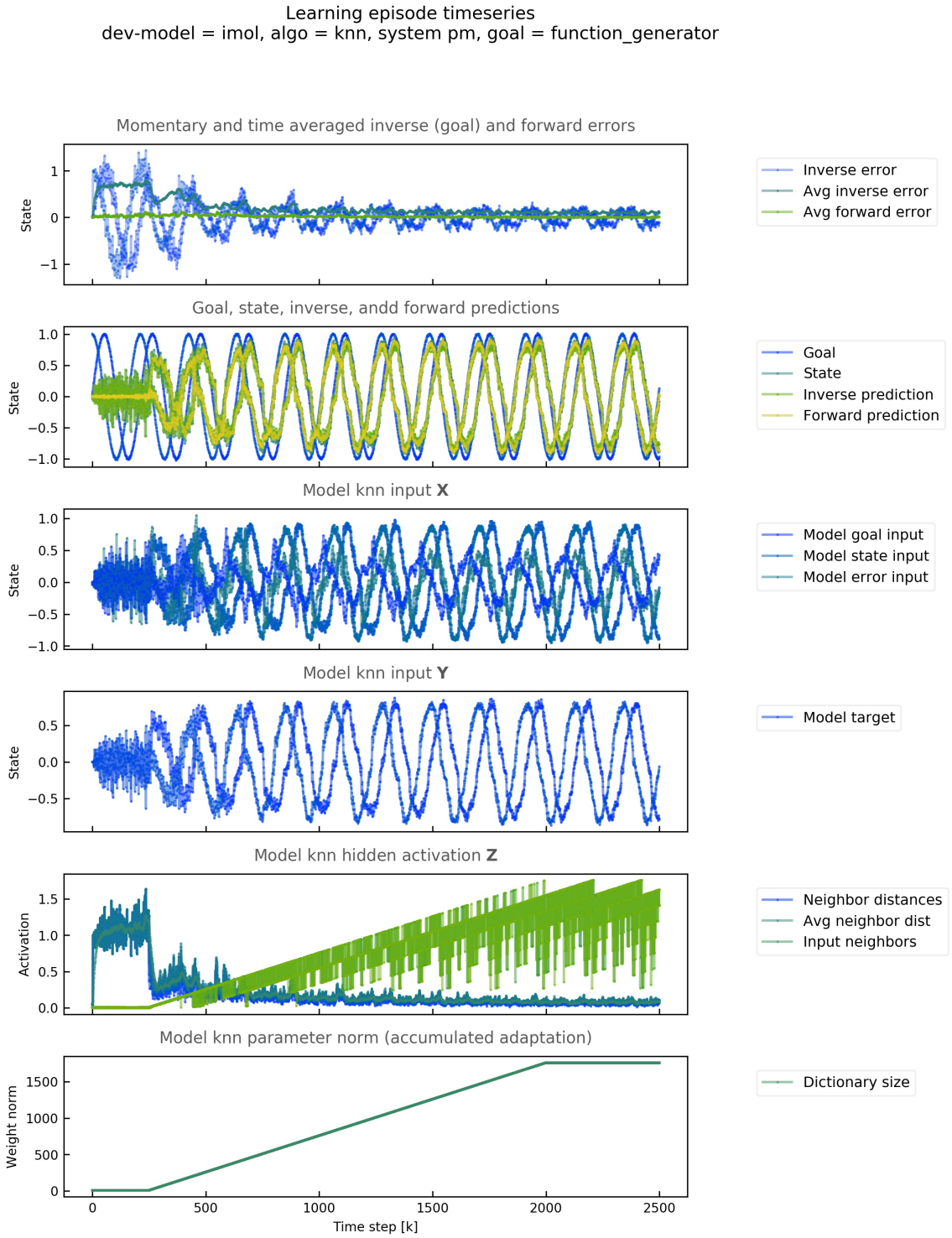



Figure 8.9.: Experiment 24-1 An internal model online learning agent learning to control a two-dimensional point mass system in the continuous goal condition using the knn low-level algorithm. The three phases of bootstrapping, learning, and testing can be read off the bottom two panels.

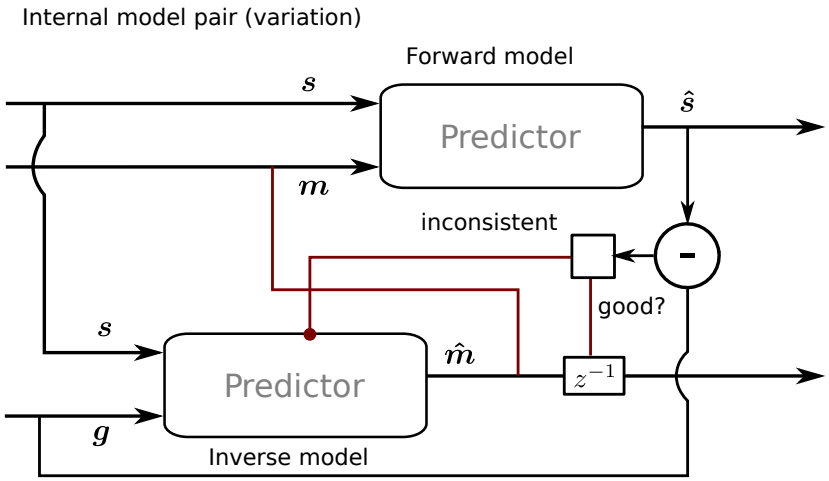



Figure 8.10.: Block diagram of an extended forward-inverse model pair. A set of internal interaction paths that make the model pair fully operational is shown in red in the picture. Interactions consists in asking the inverse model for a prediction, testing that prediction in the forward model and then keep doing that until both models *agree* that the motor prediction is adequate given current state and goal, as assessed by the forward model. If the forward model is well adapted to the current context, and simulation is fast compared to real-time demands, the simplest inverse model that suffices the overall task of the model-pair would be a uniform distribution within the motor limits.

an alternative interpretation of neural processes, commonly referred to as active inference, or predictive processing.

In predictive processing, the sensorimotor *currency* is *prediction*, througout. This serves to unify sensations and actions. In the network, there is a well-defined flow of *top-down* predictions and *bottom-up* prediction errors. Perceptual inference is the inference of some latent subspace state, based on the internal state of the full top-to-bottom prediction tree. Control is the prediction of proprioceptive states (primitive motor states). The proprioceptive system has a special role coming from the fact, that injecting a top-down proprioceptive state prediction produces physical motion as a side-effect. In addition, that motion can be expected to be locally related to raw measurements, and thus predictions, approximately by the identity. Perceptual inference in proprioceptive space necessarily leads to *active* physical motion, motivating the name active inference. At the proprioceptive level, predicting a state is the same as *producing* the state, subject to external constraints.

While this may be a confusing proposition, it is very much in agreement with the emergence of *prediction learning* in the developmental context and elsewhere, as a powerful and general tool for building unsupervised learning algorithms from supervised ones. Prediction learning data is instantly available to any neural module by wiring and learned predictors can be used to infer *synergistic* information, that is, information that is only available from the combination of two variables, a technique also known as relational learning. In entropic terms, this means leveraging the full joint entropy of two or more variables, or in probabilistic terms, to represent the joint density beyond Gaussian covariance.

The only other known implementation of an active inference model is that of (Baltieri and Buckley

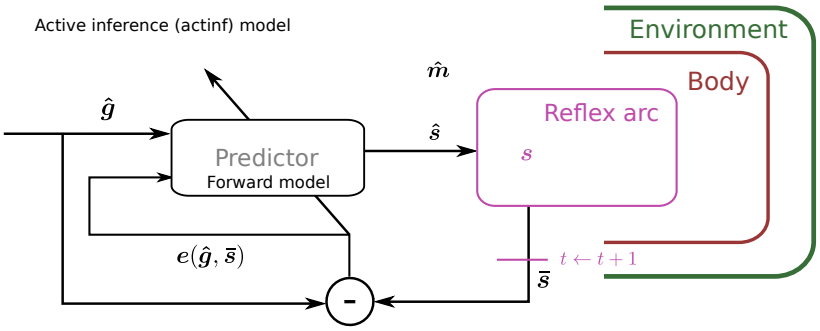



Figure 8.11.: Block diagram of the active inference model.

2017). The **actinf** model is a developmental model for bootstrapping motion skills and primitive behaviours on embodied agents, based entirely on predictive processing concepts. The only known comparabe work is (Baltieri and Buckley 2017). It is organized internally in the following way and shown graphically as a block diagram in Figure 8.11. It is assumed that some primitive sensorimotor unit P exists, which represents the proprioceptive layer. In biological discussions this unit is alternatively called a reflex arc (RA). In a robot, this corresponds to all its motor hardware components, the corresponding sensors and their driver interfaces, respectively. For example, for an angular joint actuator is can be assumed, that both the motor input unit and the angle measurement can be brought into an approximately proportional relation. Such a unit reprents one or more sensorimotor variables with the special features, that predicting them results in immediate physical action and corresponding measurement of that action. The internal mechanism of the RA unit is fundamentally outside of the agent's immediate control. The RA unit's prediction input is wired to the output of a prediction module Predictor . This module represents a low-level learning algorithm just like in the *imol* model and is repeatedly run through an incremental fit / predict sequence. Each prediction results in some physical action by wiring and an immediate prediction error. This error is used to update the model towards a target computed from the context of the actinf block by

$$\boldsymbol{Y}_{\text{fit}} = \boldsymbol{Y}_{\text{pre}} + \eta \cdot \boldsymbol{e}(\boldsymbol{g}, \boldsymbol{s})$$

with the prediction and target variable $Y$, goal $g$ and state measurement $s$, which is a greedy

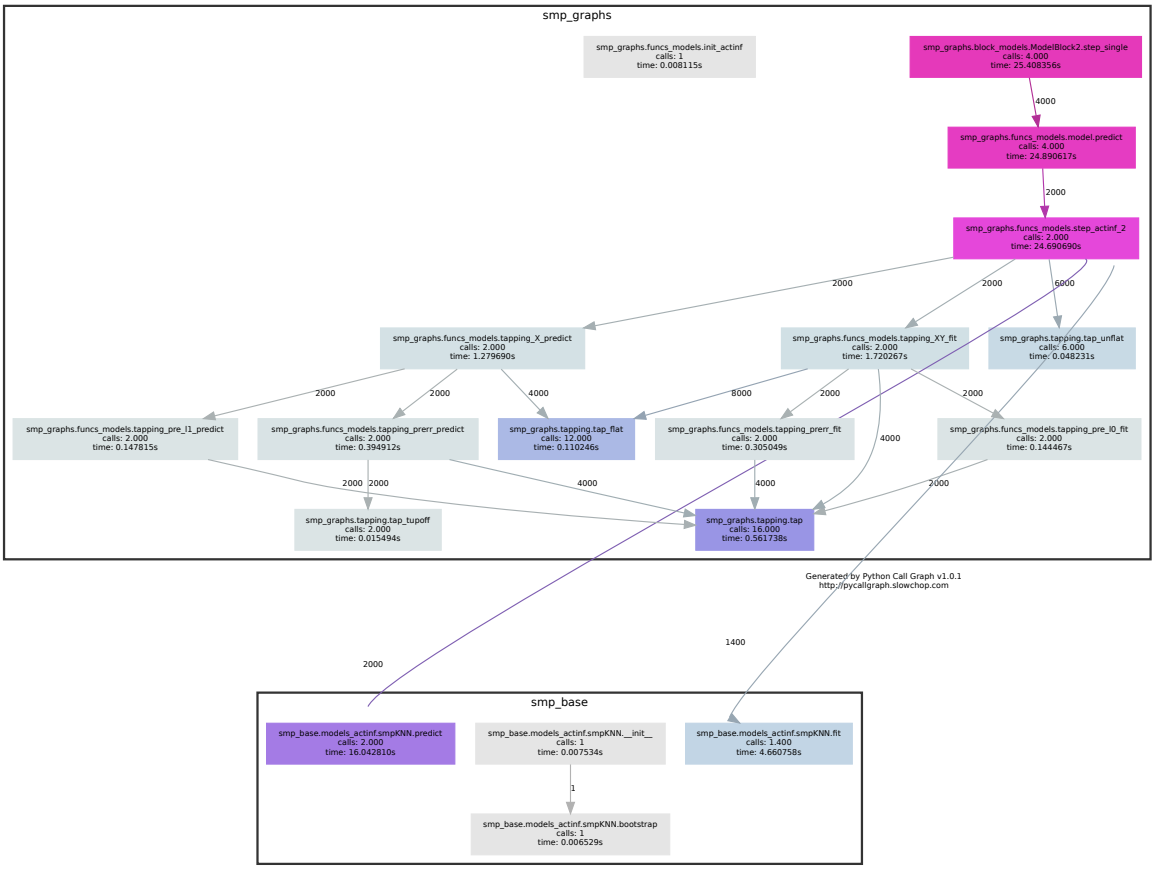



Figure 8.12.: Call graph of the active inference model.

update towards the error with respect to the top-down prediction $g$. The time index indices for these variables are the tapping of this model shown in Figure 8.13.

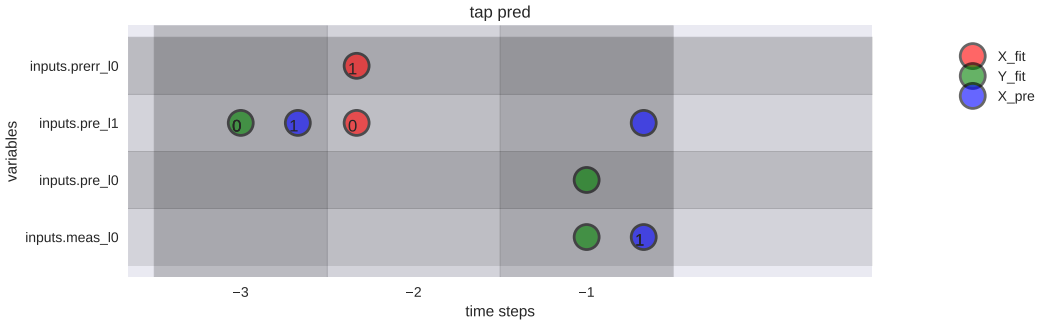



Figure 8.13.: Tapping extracted from the configuration of the smp_graphs configuration.

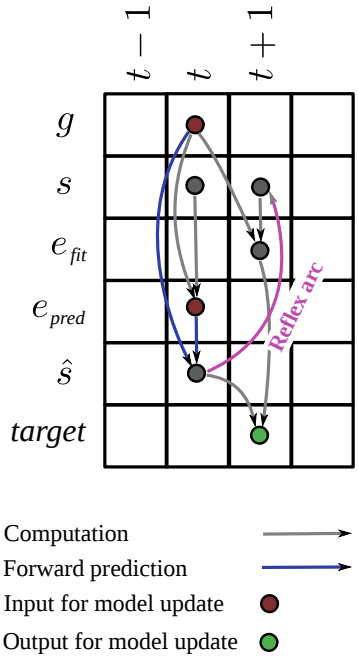



Figure 8.14.: A priori tapping derived from a principled analysis of the active inference sensorimotor loop.

**Algorithm 2** The actinf algorithm

1: $\boldsymbol{\theta}_{\text{imol}} =$ ⋯ ▷ Parameters is a configuration tree
2: $\text{model}_{\text{inv}} = [$ ▷ Populate the configuration tree with live objects
3: update_prerr_l0$_{e(g,\bar{s})}$, $\text{tap}_{\text{pred}}$, ▷ update internal state with measurement and tap state
4: $\text{smpmodel}_{\text{pred}}$, ▷ fit, predict
5: **repeat** ▷ Enter sensorimotor loop, step_actinf...
6: $\boldsymbol{s} = \text{measure}()$ ▷ obtain measurement and store it
7: **for** func in $\text{model}_{\text{pred}}$ **do** ▷ reduce the composite function stack and call each function
8: $\hat{\boldsymbol{m}} = \text{func}(\boldsymbol{s})$ ▷ that the model is comprised of by configuration
9: **end for**
10: **until** end of episode

The algorithm in **Algorithm 2** is a simplification of the imol algorithm using only a single forward-inverse hybrid predictor. The callgraph of the experiment is shown in Figure 8.12, and the tappings are shown in Figure 8.13 together with a priori tappings from a principled analysis of the actinf sensorimotor loop.

Listing 8.2: Call graph generation command with pycallgraph.

```
pycallgraph -v --exclude "*main*" \
  --include "smp_graphs.block_models.ModelBlock2.step*" \
  --include "smp_graphs.funcs_models.*actinf*" \
  --include "smp_graphs.funcs_models.*tapping*" \
  --include "smp_graphs.funcs_models.model.predict*" \
  --include "smp_graphs.tapping.*" \
  --include "smp_base.models_actinf.smpKNN.*" \
   graphviz --output-format dot --output-file dm-actinf.dot -- \
   experiment.py --no-ros --no-cache --no-saveplot --no-showplot \
     --conf conf/dm_actinf.py
```

## Experiment 25: Active inference

Similar to Experiment 23 and 24 above, this and the next one demonstrate a single developmental episode of 2000 steps of an active inference agent. The episode starts with an initial bootstrap period of 1/10th of the episode length during which the top-down prediction is applied to the low-level model which predicts from its bootstrapping state. Starting with time step 200, the model is being updated with the incoming measurements, which almost immediately brings the predicted and true state close to the goal. The goal function is a uniform random sequence of proprioceptive states. Every consecutive change of goal leads to a new mini-episode of learning for new combinations of goal prediction and local prediction error. The exploration is only partially completed when learning is stopped at time step 1600. The agent is still able to reach most goal predictions based on the existing knowledge but the remaining residual after the goal change transient is not being corrected anymore. The experimental traces are shown in similar style as previous experiments in Figure 8.15, and again discussed below.

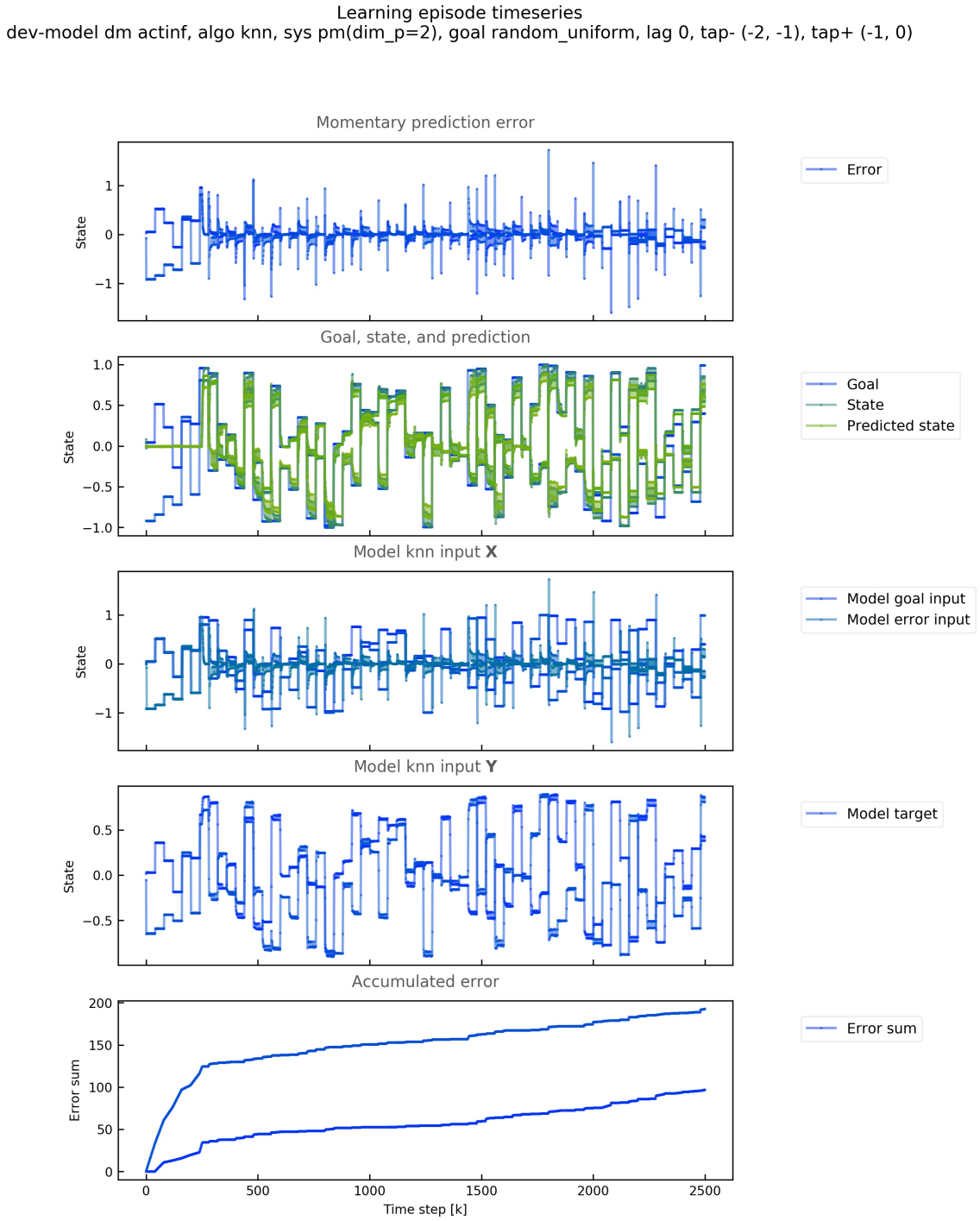



Figure 8.15.: Experiment 25-1 An active inference agent learning to control a two-dimensional point mass system in the discrete goal condition using the knn low-level algorithm. The three phases of bootstrapping, learning, and testing are most clearly seen in the second row plot where the blue goal curve appears in the beginning and end of the episode.

## Experiment 26: Active inference variation

This experiment is a goal type variation of the previous one, using a continuous target function. In the bootstrapping period nothing happens. As soon as the model update is enabled in the learning phase, the goal-state prediction error is almost instantly going to zero. In the remaining learning phase as well in the testing phase the error is close to zero. This can be seen in the first and second row of the plot in 8.16. The third and fourth row contain the traces of the low-level model inputs after tapping. The accumulated error trace in the bottom row indicates the adaption state of the model.

## 8.4. Reward-modulated Hebbian learning

The Hebbian mechanism provides a low-level description of associative learning processes that is highly autonomous, and *biologically plausible*. The degree of learning autonomy comes from the fact that learning is taking place whenever there is simultaneous activity in two neurons, regardless of their relevance to any top-down task formulation, making it a close to maximally unsupervised form of learning. Without taking energy into account, which usually happens when taking the Hebbian principle from a biological context to a purely computational one, this type of learning is unstable because of positive feedback. Thus it is clear, that the basic two factor update mechanism of

$$\boldsymbol{\Delta w} = \boldsymbol{\eta} \cdot \boldsymbol{x} \cdot \boldsymbol{y}$$

needs to be complemented with additional modulating factors $\boldsymbol{m}$ by computing

$$\boldsymbol{\Delta w} = \boldsymbol{\eta} \cdot \boldsymbol{x} \cdot \boldsymbol{y} \cdot \boldsymbol{m}$$

This is equivalent to temporal difference learning via the reward prediction hypothesis of dopamine Wolfram Schultz, Dayan, and Montague 1997. The hypothesis states that a) dopamine acts as reward indicator and b) that reward levels are predicted by the brain and the result reward prediction error is used to modulate learning via plasticity. If the reward was correctly predicted, there is no need to change behaviour. So the reward $r$ of a reinforcement learning formulation can directly be plugged into the Hebbian update equation by substituting $m$.
The algorithms proposed in (Legenstein et al. 2010; Hoerzer, Legenstein, and Maass 2012) are equivalent to the continuous actor-critic learning automaton (CACLA) in Hasselt and Wiering 2007, except for the mechanisms of reward and value function learning. In CACLA, the state-action value function is approximated over multiple episodes via fitted Q-learning. Value function learning serves to interpolate from an initial reward function that is potentially very sparse in the state-action space, to a smooth and reward prediction (the value) that is dense in the states and that can be used for model updates at every time step. The question about the dependence among A, B, E and $P$ can also be posed in terms of value function reusability.
It is plausible to assume that a set of relevant value functions are given as genetic priors in the biological situation and in many cases of sensorimotor skill learning problems, smooth reward

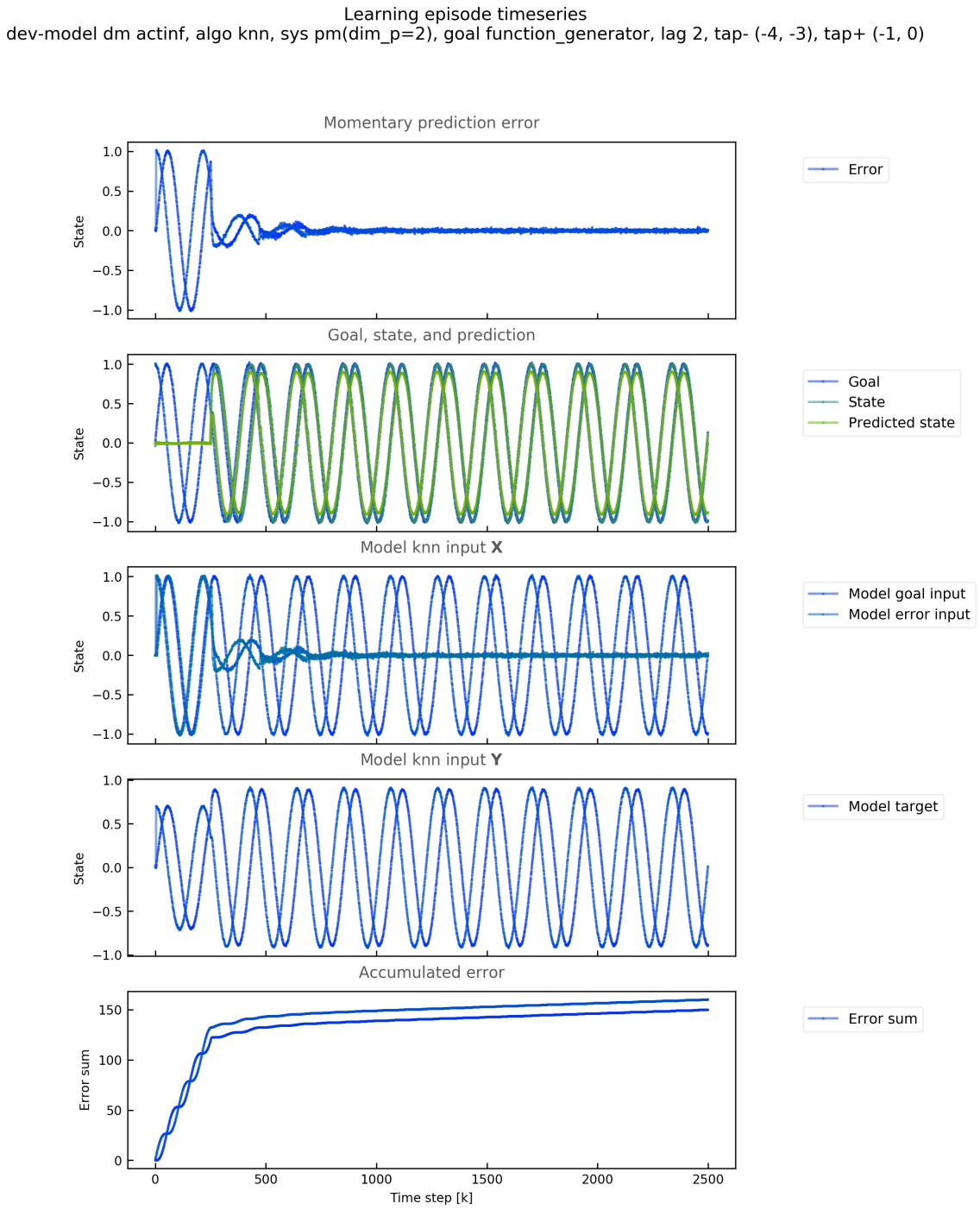



Figure 8.16.: Experiment 26-1: An active inference agent learning to control a two-dimensional point mass system in the continuous goal condition using the knn low-level algorithm. At the onset of learning the effect is almost immediate (second row, green curve). The testing phase is visually indistinguishable from learning.

functions can be given precisely or approximately. In addition, the prediction learning trick can also be leveraged for the Hebbian model, providing smooth rewards for free simply from the time contiguous prediction errors, which links back to dopamine prediction. In the following experiments, only fixed reward functions are used. It can be expected, that replacing these prior rewards with value functions learned specifically for a given sensorimotor context will only improve the learning process.

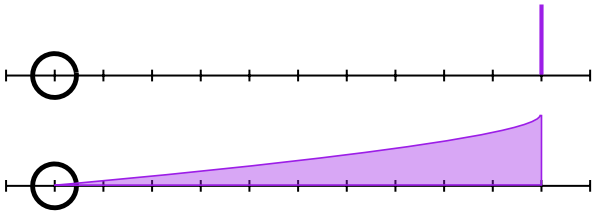


Figure 8.17.: Illustration of the interpolating effect of value learning with respect to initially sparse rewards.

Our scenario consists of a robot, an environment and a neural control circuit. The robot we consider here is the Sphero, the environment consist of the transformation laws describing how new sensor states are computed from a motor command, and the neural circuit is a reservoir network. It consists of an input layer, a single hidden layer of recurrently connected neurons, and a linear output layer referred to as *readouts*.
Reservoirs are most often trained with batches of supervised training sets (Lukoševičius and Jaeger 2009). Here, instead, we employ incremental updates with a Hebbian learning rule. The learning rule is modulated by a performance measure defined on sensor states. We assign problem specific performance measures, in this case for example the negative quadratic distance to an externally provided target value. The learning task is to invert the robot-environment coupling to find the motor command which generates a sensor state representing good performance.
The inverse model is realized as a function $\boldsymbol{y} = f(\boldsymbol{u}, \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{W}^{\text{out}})$ mapping sensory inputs $\boldsymbol{u} \in \mathbb{R}^n$ to motor outputs $\boldsymbol{y} \in \mathbb{R}^m$ with $n$ the sensor dimension and $m$ the motor dimension, $\boldsymbol{x} \in \mathbb{R}^N$ is a hidden state with dimension $N >> n$. The hidden state is both driven by the sensory input and recurring upon itself via connection strengths drawn randomly once at the beginning of the experiment. The motor output $\boldsymbol{y}$ is a linear combination of the hidden state with weights $\boldsymbol{W}^{\text{out}}$. We realize the function $f$ as a reservoir.
Exploiting the universal modelling properties of reservoir networks, the task is reduced to finding parameters $\boldsymbol{W}^{\text{out}}$ that implicitly encode the inverse model. Without explicitly computing the performance gradient with respect to the parameters, we update the weights with a Hebbian rule which is modulated by a third factor. This factor is a binary indicator of recent improvements in performance. The accumulated weight changes reinforce successful actions (post-synaptic activation) for corresponding hidden states (pre-synaptic activation) when the reward signal is non-zero. It amplifies the correlation between the rewards generated by an exploration signal and the pre- and postsynaptic states (Hoerzer, Legenstein, and Maass 2012) such that states which yield reward are made more likely to occur.
The reason for choosing this type of learning is, that no explicit target for the motor signal. Conventional reservoir training assumes the existence of a supervised training set in terms of

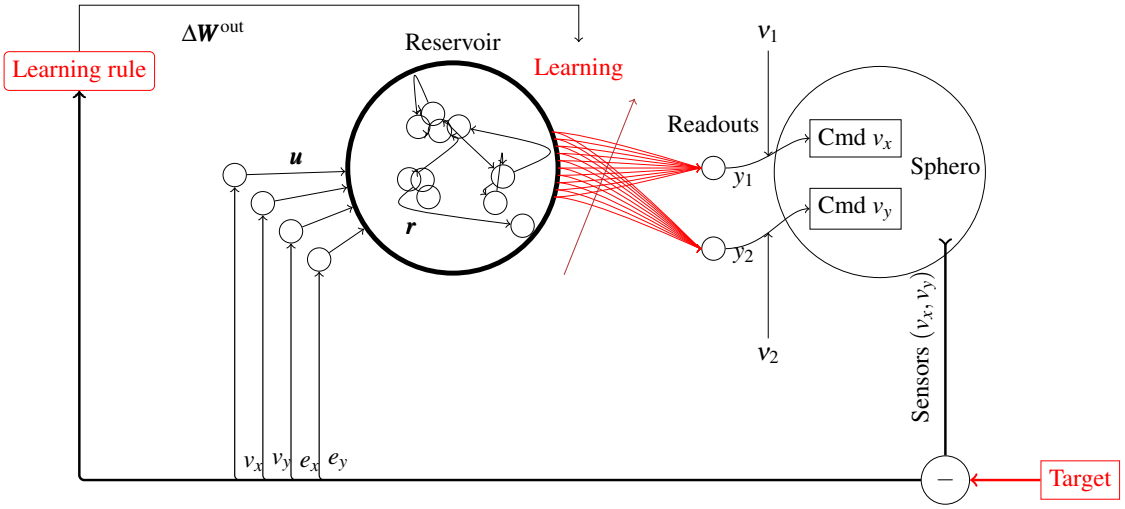
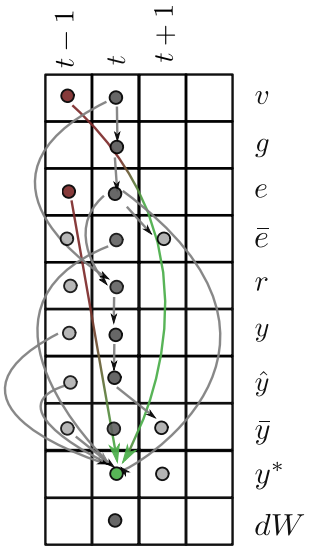



Figure 8.18.: Graphical representation of the learning algorithm. The thick circle labeled Reservoir implements Eqs. 8.1,8.2 and 8.3, the red bundle of arrows and neurons $y_1$ and $y_2$ correspond to Eq. 8.4. After that, noise $\nu_1$ and $\nu_2$ are added and sent to Sphero's control input (Eq. 8.5). The red box "Learning rule" contains both Eqs. 8.6 and 8.7. The boxes labelled "Cmd" also contain the output scaling factor $\text{gain}_{\text{out}}$ which is specific and usually constant for a given robot. The variables $v_{x,y}$ and $e_{x,y}$ are the measured velocity and velocity errors respectively.




$v_t, e_t \rightarrow y^*_{t+1}$ is the fit

the Hebbian update

$$dW = r \cdot (\hat{y} - \bar{y}) \cdot (e - \bar{e})$$

implies a target that can be given as

$$y^* = y + r \cdot dy \cdot de$$

Figure 8.19.: Illustration of the interpolating effect of value learning with respect to initially sparse rewards.

input and output pairs but here the target is only given indirectly. The real output target has to be discovered by the learner through exploration. The function $f$ is realized in the following way by a reservoir. The hidden network state evolves according to

$$\triangle \boldsymbol{x}_t = \lambda \boldsymbol{W}^{\text{res}} \boldsymbol{r}_t + \boldsymbol{W}^{\text{in}} \boldsymbol{u}_t + \boldsymbol{W}^{\text{bias}} 1 \tag{8.1}$$

$$\boldsymbol{x}_{t+1} = (1-\tau)\boldsymbol{x}_t + \tau \triangle \boldsymbol{x}_t \tag{8.2}$$

$$\boldsymbol{r}_{t+1} = \tanh(\boldsymbol{x}_{t+1}) + \nu_{\text{state}} \tag{8.3}$$

The matrices $\boldsymbol{W}^{\text{res}}, \boldsymbol{W}^{\text{in}}, \boldsymbol{W}^{\text{bias}}$ are the $N \times N$ reservoir, $N \times n$ input, and $N \times 1$ bias matrices respectively. We use a reservoir size of $N = 500$, and input dimension $n = 4$. The scalar $\lambda = 0.99$ is a scaling factor to effect a desired spectral radius for the reservoir matrix. The connection probability for the reservoir matrix $\boldsymbol{W}^{\text{res}}$ is controlled by parameter $p = 0.1$. This means, we generate the matrix with sparse non-zero entries of density $p$. The non-zero entries themselves are drawn from a standard normal distribution. Then the matrix is rescaled to the given spectral radius $\lambda$. The state noise $\nu_{state}$ is uniformly distributed with limits -0.02 and 0.02 and is used as a regularizer. The network outputs are computed as

$$\boldsymbol{y} = \boldsymbol{W}^{\text{out}} \boldsymbol{r} \tag{8.4}$$

At this stage white Gaussian noise $\boldsymbol{\nu}$ is added to the output $\boldsymbol{y}$ yielding

$$\hat{\boldsymbol{y}} = \boldsymbol{y} + \boldsymbol{\nu} \tag{8.5}$$

This is the final stage motor signal before it is sent to the actuators.

### 8.4.1. Performance measure and learning rule

We designate the measure of the current performance of the network with $P$. It depends on the motor output $k$ time-steps in the past. Here, we mostly use the negative squared error with respect to an externally imposed target such as $P_i = -(u_i - \text{target}_i)^2$ for a given sensory input $i$ or the sum $P = -\sum_i (u_i - \text{target}_i)^2$. A low-pass filtered version $\bar{P}_t = \alpha P_{t-1} + (1-\alpha) P_t$ is also maintained with $\alpha = 0.8$. The modulator signal is derived as an approximation of the performance derivative from $P_t$ and $\bar{P}_t$ via

$$M_t = \begin{cases} 1 & \text{if } P_t > \bar{P}_t \\ 0 & \text{otherwise} \end{cases} \tag{8.6}$$

and the weight update then is

$$\Delta \boldsymbol{W}^{\text{out}}_{i,t} = \eta_{i,t} \boldsymbol{r}_{t-k} (y_{i,t-k} - \bar{y}_{i,t-k}) M_t \tag{8.7}$$

with $\bar{y}_{i,t}$ being a low-pass filtered version of $y_{i,t}$ analogously to $\bar{P}$. We use a time-dependent learning rate $\eta_t$ with a half-time on the order of 1000 time steps. Also, we apply soft weight-bounding to avoid run-away solutions for the output weight vector $\boldsymbol{W}^{\text{out}}$. The weight bounding

is an additional multiplicative term in the update rule, which throttles the learning rate to zero if the norm of the weight vector comes close to an empirically determined threshold. Note that the standard Hebbian terms are indexed with $t-k$ whereas the modulator refers to the current time. The variable $k$ is the sensorimotor delay which needs to be determined either through knowledge of the system or empirically. In the latter case this can be done using cross-correlation analysis. A graphical representation of the algorithm is displayed in Figure 8.18 and a pseudo code form is given in

**Algorithm 3** The standard EH-rule

1: $N \leftarrow$ state dimensionality, $H \leftarrow 1$ eligibility window size
2: **repeat** ▷ forever
3: exploration step $i$
4: $\Delta w = \eta \cdot h_k \cdot r_{t-k} \cdot \frac{\Delta y}{\Delta t}_{t-k} \cdot M$ ▷ Apply learning rule
5: $\Delta \text{cw}_k = \sum_{j=[1,\ldots,N]} \Delta w_{k,j}$ ▷ Accumulate weight changes for learning step $i$
6: $w = w + \Delta w$
7: **until** end of episode

## Experiment 27: Reward-modulated learning

Similar again to previous experiments in this chapter, this ones illustrates a learning episode of an exploratory Hebbian agent. In the first variation, the task for the agent is to follow a discrete sequence uniformly random goals. After washout where the episode starts with a zero output model, the learner is slower to pick up on the target signal, as compared with the previous two models. This is to be expected from the fact that the learning rule only uses a binary reward signal. The reward is one if the current error is less than the predicted error (differential Hebbian learning). The binary reward contains less information than an error with sign and magnitude. In addition, the time constant of the recurrent hidden activation is not well adjusted to the goal condition of instantaneous jumps. As can be seen in the botton row plot in Figure 8.20, the model is still adapting when the testing phase starts at time step 2000. Nonetheless, the overall performance is adequate during the testing phase at the end of the episode. This is visible in the second row of the figure where the green state curve covers the goal curve to large extent indicating that a correct model was acquired.

## Experiment 28: Reward-modulated learning variation

This experiment represents a variation of the previous one using a continuous goal function. In this condition, the approach toward the target is slightly slower for the exploratory Hebbian learner than for previous models. The reason is again a different kind of error signal. The goal state still met adequately a learning transient that finishes before the testing phase starts. This can be seen in the bottom row plot of Figure 8.21 as a sharp bend at, and flat continuation of the curve after

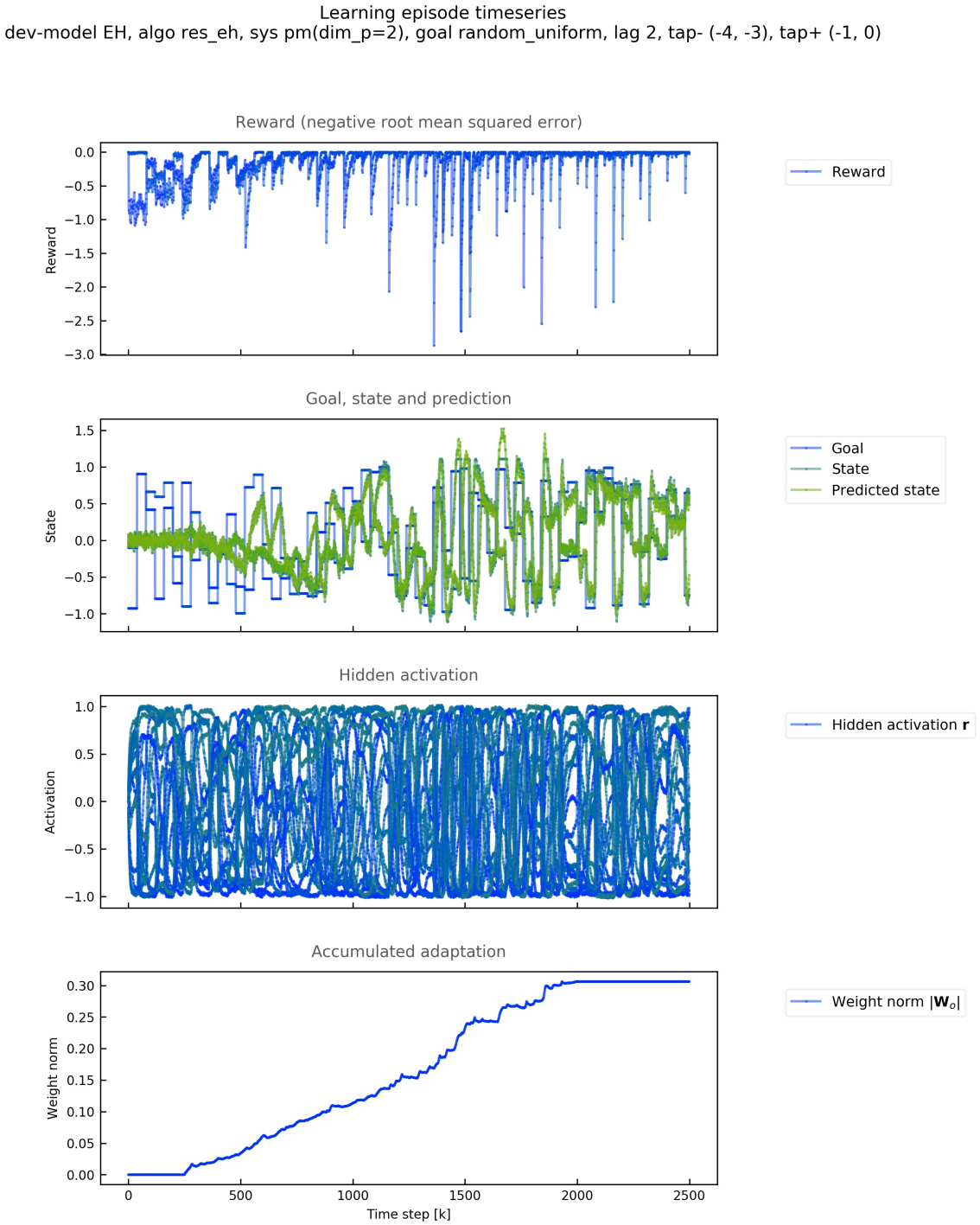



Figure 8.20.: Experiment 27-1: An exploratory Hebbian agent learning to control a two-dimensional point mass system in the discrete goal condition using the reservoir-EH low-level algorithm. The three phases of bootstrapping, learning, and testing are most clearly seen in the botton row plot of the output weight norm.

half of the episode's duration. Correspondingly, the overlap of the blue and green curves in the second row plot confirms sufficient testing performance and correctly learned model.

### 8.4.2. Tappings revisited as eligibility traces

Rate coded correlational learning models are usually discussed assuming zero or unit delay between relevant stimuli. Applying correlational learning on real systems with unknown delays can be handled by the tapping framework, as for other developmental model. An interesting side results comes from applying eligibility traces to the problem. An eligibility trace is the product of a variable's past with an exponentially decaying window function. The weight is coding for how much a past action is eligible to taking the credit for the current reward. An extension of the EH algorithm is derived, called EH extended (EHE) and shown in **Algorithm 4**, which uses a contiguous window of fixed size to accomplish the learning task as in the original EH rule and which allows to extract temporal offsets which accumulate large rewards. These offsets can be used as a tapping because accumulated reward indicates task relevance. A more detailed presentation is given in section C.3 of the Appendix.
The episodes of developmental learning processes presented in this chapter for three different types of models have only been discussed qualitatively. The experiments show that each model can quickly acquire the required skills which are not available to the agent at the beginning of each episode. The concluding experiment of this chapter serves to put this on quantitative grounds.

## Experiment 29: Model comparison

This experiment provides statistics on the performance of each model proposed. A uniform random strategy is provided as a baseline for comparison. The root mean squared error between the goal and the state is shown in the box plot of Figure 8.22 for each configuration and averaged over 100 runs. Results are shown for two different goal conditions, discrete uniform random goals and the continuous sinusoidal one. All three models perform consistently at similar order of magnitude and better than the baseline. The only exception of the exploratory Hebbian model in the discrete goal condition, where some outliers are generated that performing arbitrarily worse. This is due to the hyperparameters of the underlying low-level model, which have not been optimized for the high-frequency content of the discrete goal condition, where large errors can destabilize the learning process. In the the continuous goal condition it reliably outperforms the IMOL model.

## 8.5. Results

In this chapter, three different developmental models have been presented. The first one is based on the well-established concept of forward / inverse model pairs in sensorimotor theory and is called **IMOL**. It provides a state of the art baseline for comparison of the novel models. These are an active inference model called **actinf**, and the reward modulated Hebbian model called **EH**. The perspectives highlighted in the presentation included the domain-centered modeling of developmental processes, the low-level machine learning aspects, and an approach to systematic

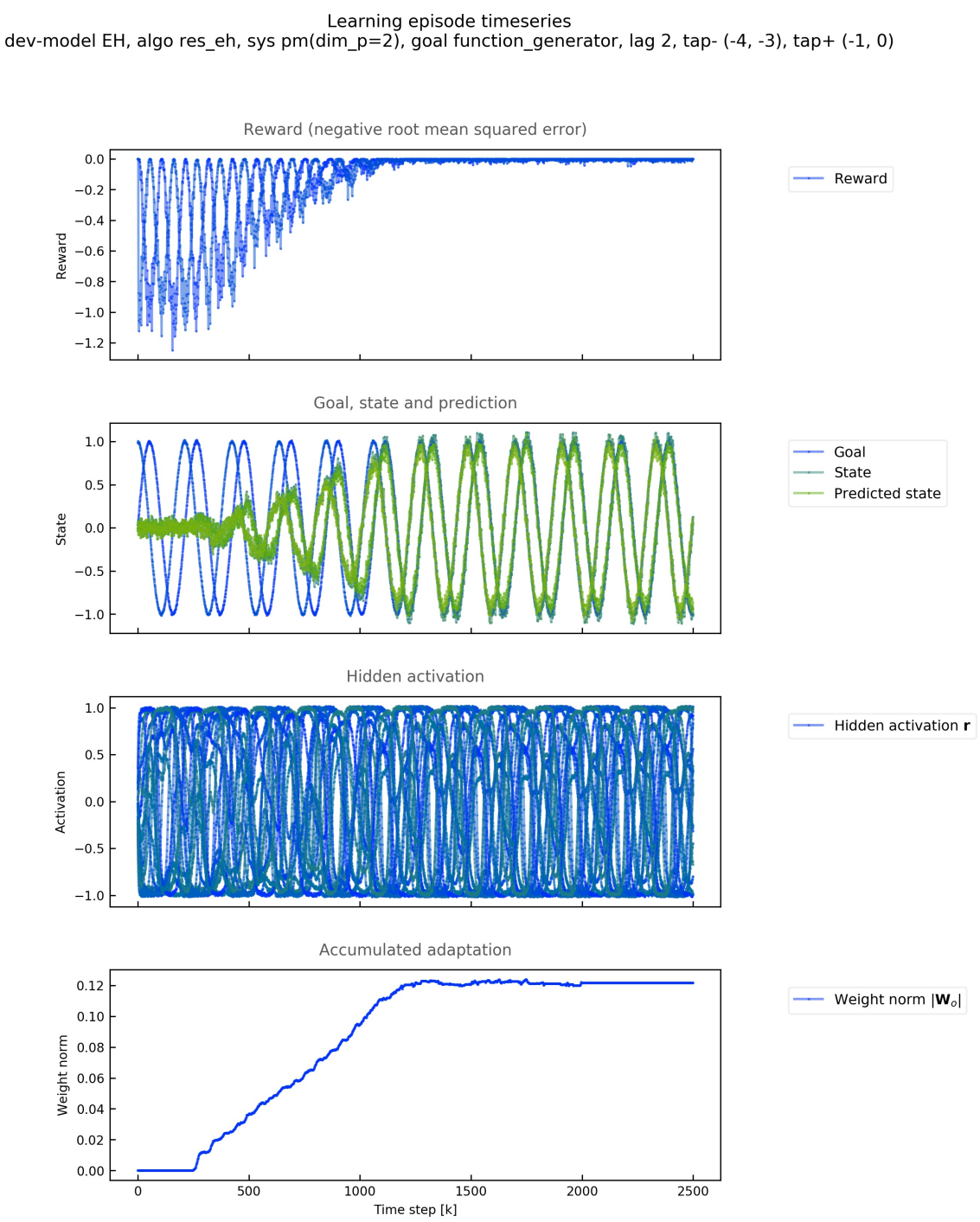



Figure 8.21.: Experiment 28-1: An exploratory Hebbian agent learning to control a two-dimensional point mass system in the continuous goal condition using the reservoir-EH low-level algorithm. In this condition, learning is converged after half of the episode and stable behaviour persists during the testing phase.

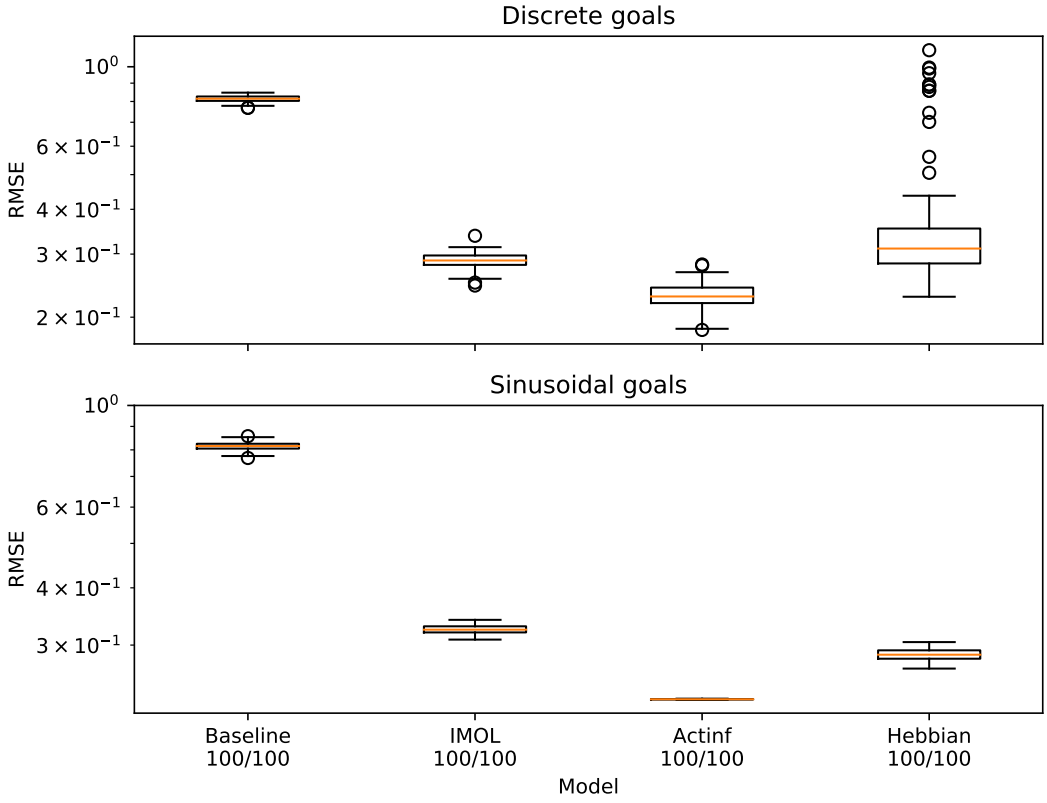



Figure 8.22.: Experiment 29 is a comparison of the three models proposed earlier. These are the IMOL, actinf and exploratory Hebbian models. For comparison a random strategy baseline is provided. All three models perform reliably on a similar order of magnitude and better than the baseline with the exception of outliers in the exploratory Hebbian discrete goal condition, due to the mismatch of goal condition and hyperparameters discussed in the main text.

implementation and variation of the resulting embodied agent experiments within the smp_graphs library. All experiments in this chapter make use of the tappings introduced earlier, which provide an explicit representation of dependency in information flows in embodied agents. The main results of the chapter are the three models which are set up on the basis of a sound self-exploration and shown to reliably achieve a sufficient degree of control for robot motion. They provide a set of minimal yet extensible solutions for the initial problem of how a robot can learn about its body from scratch.

# Part III.

# Conclusion

# 9. Discussion and outlook

This part concludes the thesis, starting with a summary and followed by a discussion and outlook on future work. In part I, the thesis was motivated and set up to investigate a problem posed as

> How can a robot learn about its body from scratch?

The question aims at general solutions to a set of specific robot motion learning problems. Within bio-inspired approaches a developmental robotics perspective was adopted. This was done because developmental learning itself represents such a general solution. The necessity of a sensorimotor account of behaviour emphasizes aspects such as real time interaction, active information shaping, and goal generation in robot learning that are neglected in other approaches.
The main body of the thesis is part II, consisting of three chapters. In chapter 6 the sensorimotor framework is erected. A methodology of systematic iterative refinement of experiments was pursued, which is related to ideas of a discovery science (Biswal et al. 2010). To be able to generate variations systematically, the graph based language *smp_graphs* was designed for specifying sensorimotor learning experiments. The graphical description is converted into an executable computation graph to run each experiment. The framework of the thesis along with configuration language is based on state of the art sensorimotor theory as well as existing approaches to graphical modeling and systematic variation in computational experiments in other contexts. In the second half of chapter 6, the framework is used to formulate a succession of agents that starts with a random baseline and each step increasing the agent's capabilities. The resulting behaviours are quantified and compared using a provisional aggregate measurement that puts the mean squared error side by side with statistical distances and an external "survival" reference. This is done in an effort to gauge dimensionless internal quantities to adequacy in the external frame of reference.
The problem is then considered as two different developmental phases, self-exploration and skill acquisition. The treatment of self-exploration in chapter 7 is meant to answer questions about the prerequisites of skill acquisition. This is based on the idea that the *self* can be distinguished from *external* influences as a characteristic invariant in sensorimotor data. If this is the case, it is beneficial to account for this in the developmental schedule and learn about the self before interaction skills with the outside are learned. This allows to disentangle effects which would be much harder to discern using a monolithic approach. Two main contributions are presented in the chapter. A graphical approach called *tappings* is proposed for representing the generating information footprint of the current agent state within multivariate sensorimotor timeseries. Tappings are then used to analyze and compare different learning schemes commonly found in the literature. The purely conceptual framework is then extended with a method to learn the input graph structure and augmented it with continuous valued relevance weights. This is done by decomposing multivariate sensorimotor data into information based dependency estimates using

a scanning technique called *infoscans*. An algorithm is outlined and its application illustrated in a corresponding set of experiments.
In chapter 8, skill acquisition is considered on the basis of the results of the preceding chapter. The presentation focusses on the variation of learning algorithms in online learning scenarios. The main contribution of the chapter is a functional decomposition of developmental models for acquiring primitive motion skills on embodied agents. The decomposition comprises to large extent of subordinate adaptive internal model blocks, which can be configured through their tapping to acquire different kinds of functions, in the context of the enclosing developmental process. This provides functional invariance based on adaption to the impinging sensorimotor statistics. Three different instances of the general model are developed within the chapter, an online learning forward-inverse model pair (imol), an active inference model driven by prediction error (actinf), and a reward-based correlational learner (EH). Each model's behaviour is illustrated in two experiments each, with different goal conditions in each experiment. These experiments highlight that very short learning transients can be achieved given appropriate sensory feedback is established through self-exploration. All three models are then evaluated for robustness via their respective error statistics on a sample size of 100 randomized runs of the preceding experiment configurations. All models proposed evaluate favorably against the baseline at robust levels, disregarding occasional stability issues. With these results, a minimal but complete account of a developmental schedule for effective motion skill learning for embodied robots has been accomplished.

## 9.1. Future work

The thesis is developed in a minimal spirit making it necessary to concede to a lot of abstraction and scaffolding. Some of the more salient items needing to be improved along these lines are given now.
The developmental phases proposed above need to be integrated further into a single experiment. This requires a closed-loop function to modulate the activity of each phase in an appropriate way. The possibility has been sketched previously to achieve such a modulation by filtering the error signal at two different time scales and comparing these two quantities. Different ratios can be classified into four discrete states. With both errors small, nothing needs to be changed. If the fast error is smaller than the slow error, this indicates improvement and current learning can proceed. The fast error being larger than the slow one indicates degradation on the other hand. Both errors being large means something is seriously wrong and a major reconfiguration is needed, for example learning a new model. A more advanced version will also consider the error derivatives.
The experiments presented in the main chapters are kept simple for readability in terms of the state dimensions and coupling configurations. The framework has been evaluated extensively in additional experiments not documented here with two exceptions that can be found in chapter C of the Appendix. The chapter includes links to videos, a hyperparameter optimization experiment on the point mass system, an investigation of eligibility traces in reward modulated learning, and an internal model learning experiment on a simulated quadrotor. These studies examine to varying extent more degrees of freedom, high-dimensional visual input, and different simulated and real

robots. Simulated systems include the explauto (Moulin-Frier, Rouanet, and Oudeyer 2014) arm, LPZRobots' (Der and Martius 2012) barrel. Real robots that were used include the Sphero, a Turtlebot, an RC model car, different quadrotor simulations, and the Nao robot. Robots can be connected using ROS from within the smp_graphs software and all three developmental model variants of the skill acquisition chapter have been tested on these systems. Taken together this provides routes to be followed up for demonstrating the scalability of the current approach.
The proposed framework can be used for systematic model search, also known as model selection or hyperparameter optimization. A large number of approaches exist to do this in a principled way that improves on grid- and random search techniques. Two major families are evolutionary methods, for example Covariance matrix adaptation evolutionary strategies (CMA-ES) and Bayesian hyperparameter optimization algorithms, for example hyperopt. The application of hyperopt is shown in the experiment of section C.2.
On a more direct note, the smp_graphs library provided a good fabric for the work untertaken here, but some limitations are already becoming evident. While the language is expressive enough to describe the development models in question, the programs still become quite long and confusing for many scenarios. Thus the syntax needs to be refined to become much more compact. Also it is evident the language's dynamic self-manipulation capabilities need to be improved as a prerequisite for procedural generation of graphs and for the seamless integration of growth functions.
The perspective of growth is present as a latent thought in the thesis. Long-term autonomy through ongoing adaptation will require growth on the internal model level. Existing growth based learning algorithms are mostly extensions of the growing neural gas model (Fritzke 1995), a variant of the self-organizing map. Growth is well supported by the predictive processing framework, where models are thought of as layers stacked on top of each other with a successively larger scope of integration along the bottom to top direction. The proprioceptive model of the skill acquisition chapter includes references to this interpretation, viewing a model's goal input as a top-down prediction from one level up. A complementary approach is suggested by residual deep neural networks. In the original proposal (He et al. 2016), the input is passed to every layer in a network by a shortcut connection. At the same time each layer receives the prediction error from the preceeding layer. This requires each layer to predict successively less of the original input and use the adaptive resources for explaining the residual through accumulated integration, thus the name residual network. The original ResNet was proposed as a static structure but it can be modified to grow successive layers dynamically based on residual error levels.

## 9.2. Closing notes

The vision at the start of this thesis consisted of the idea of *sensorimotor primitives* which is visible as shorthand term *smp* used in reference to the software. The primitives were to be represented by neural networks and trained through biologically plausible learning algorithms. The model would allow an embodied agent to bootstrap self-control and coordinated behaviour up to the point of the stability required for avoiding self-destruction. The model should be general enough to work on different robots and would be called Learning in a Box. The suspicion existed that a small set of identifiable primitives would emerge from rerunning the model on different systems, producing a catalogue of readily reusable adaptive modules.

The final state of the project as presented here is not too far from the original vision, although many larger and smaller aspects had to be discarded and many new ideas fended off. A picture that emerged while this work was being finished is that of a *pile*. Such a pile is shown in Figure 9.1 on the right hand side next to a scaffold like structure on the left. Both structures represent an agent. The pile results from the idea of piling up behavioural modules on each other. Each module is like an amorphous rubber sheet covering some part of the sensorimotor data space. Such a sheet moving into unoccupied regions of the space is a metaphor for adaptation to any kind of information environment. The global behaviour is generated by the joint acitivity of sheets that make up the pile and will be very robust against removal or deformation of sheets below.

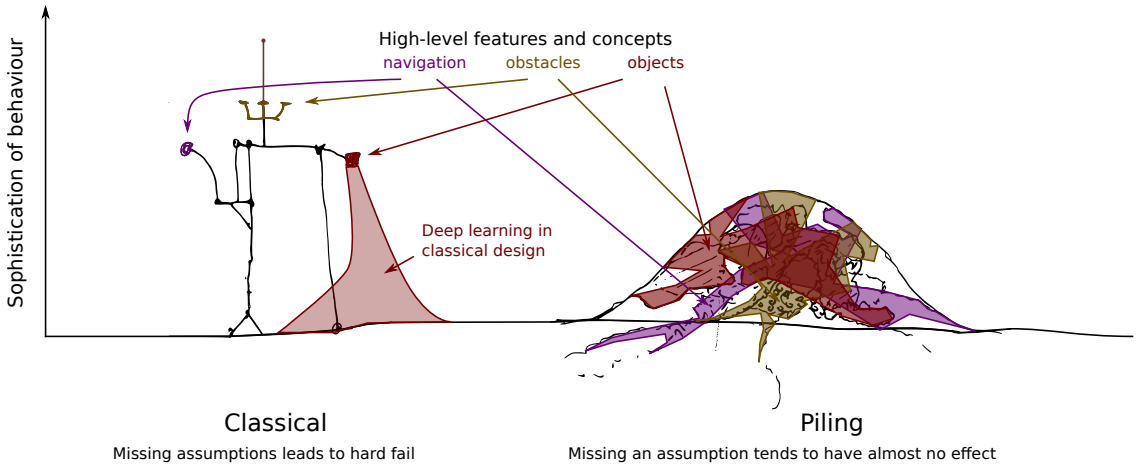



Figure 9.1.: Two different agent designs, one shown as a pile of functions on the right hand side, and the other shown as a scaffold like structure on the left. The desired sophistication of behaviour is given by the y-axis. The question is what happens when a priori assumptions are violated during the lifecycle of an agent. The picture suggests the piling of function as an alternative design principle that could prove more robust and fail better in such a case.

## 9.3. Acknowledgements

The conclusion of this projects was only possible through infinite support by numerous individuals and organisations. These are my parents, my family, my mentors and supervisors Verena Hafner, Beate Meffert, Frank Winkler, all of my colleagues and students over the years, in particular Christian Blum, Guido Schillaci, Damien Drix, Aleke Nolte, Andreas Gerken, Benjamin Schlotter, and Florens Greßner, the NaoTH team, the Department of Computer Science at the HU Berlin, the Humboldt-Universität zu Berlin, the Deutsche Forschungsgesellschaft (DFG), neurocat GmbH, my friends, and some more that I have missed to name.

# Part IV.

# References

# Bibliography


Adams, Rick A, Stewart Shipp, and Karl J Friston (2013). "Predictions not commands: active inference in the motor system". In: *Brain Structure and Function* 218.3, pp. 611–643.

Alain, Guillaume and Yoshua Bengio (Oct. 2016). "Understanding intermediate layers using linear classifier probes". In: *arXiv:1610.01644 [cs, stat]*. arXiv: 1610.01644. URL: http://arxiv.org/abs/1610.01644 (visited on 02/02/2018).

Asada, M. et al. (May 2009). "Cognitive Developmental Robotics: A Survey". en. In: *IEEE Transactions on Autonomous Mental Development* 1.1, pp. 12–34. ISSN: 1943-0604, 1943-0612. DOI: 10.1109/TAMD.2009.2021702. URL: http://ieeexplore.ieee.org/document/4895715/ (visited on 06/11/2018).

Ashby, W. Ross (1952). *Design for a Brain*. Chapman and Hall.

Baltieri, Manuel and Christopher L Buckley (2017). "An active inference implementation of phototaxis". In: *arXiv preprint arXiv:1707.01806*.

Barto, A. G. (1995). "Adaptive critics and the basal ganglia". In: *Models of information processing in the basal ganglia*. Ed. by J. C. Houk, J. L. Davis, and D. G. Beiser. Cambridge, MA, USA: MIT Press, pp. 215–232.

Benureau, Fabien (2015). "Self Exploration of Sensorimotor Spaces in Robots." PhD Thesis. Université de Bordeaux.

Benureau, Fabien, Paul Fudal, and Pierre-Yves Oudeyer (Oct. 2014). "Reusing motor commands to learn object interaction". en. In: IEEE, pp. 343–350. ISBN: 978-1-4799-7540-2. DOI: 10.1109/DEVLRN.2014.6983004. URL: http://ieeexplore.ieee.org/document/6983004/ (visited on 06/11/2018).

Bergstra, James S et al. (2011). "Algorithms for hyper-parameter optimization". In: *Advances in neural information processing systems*, pp. 2546–2554.

Bergstra, James, Daniel Yamins, and David Cox (2013). "Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures". In: *Proceedings of The 30th International Conference on Machine Learning*, pp. 115–123.

Bernshteĭn, N. A. (1967). *The co-ordination and regulation of movements*. English. Pergamon Press. URL: http://books.google.com/books?id=F9dqAAAAMAAJ.

Berthold, Oswald (Dec. 2009). *simctrl - crrcsim simulator control shell*. URL: http://www.informatik.hu-berlin.de/%20oberthol/QK/.

– (2011). *An Approach to UAV Controller Prototyping with Linux*. en. Student research project report. Humboldt-Universität zu Berlin, Institut für Informatik, p. 39. URL: https://www2.informatik.hu-berlin.de/~bertolos/QK/Studienarbeit.pdf (visited on 06/10/2018).

– (2015). *Correlational learning with unknown delay*. Tech. rep. Humboldt-Universität zu Berlin, Institut für Informatik.

– (2018a). *smp_base*. URL: https://github.com/x75/smp_base.

– (Apr. 2018b). *smp_graphs*. URL: https://github.com/x75/smp_graphs.

Berthold, Oswald and Verena Hafner (Apr. 2017). "Tapping the sensorimotor trajectory". In: *arXiv:1704.07622 [cs]*. arXiv: 1704.07622. URL: http://arxiv.org/abs/1704.07622 (visited on 04/04/2018).

Berthold, Oswald and Verena V. Hafner (2013a). "Neural sensorimotor primitives for vision-controlled flying robots". Published: IROS'13 Workshop Closed-loop vision controlled MAVs. URL: http://rpg.ifi.uzh.ch/docs/IROS13workshop/Berthold.pdf.

– (2013b). *Reward-based Hebbian learning of robotic motor primitives using random networks*. Tech. rep.

– (2013c). "Unsupervised learning of camera exposure control using randomly connected neural networks". In: vol. 1. Compiegne, France.

– (2014). "Unsupervised Learning of Sensory Primitives from Optical Flow Fields". In: *From Animals to Animats 13: 13th International Conference on Simulation of Adaptive Behavior, SAB 2014, Castellón, Spain, July 22-25, 2014. Proceedings*. Ed. by Angel P. del Pobil et al. Cham: Springer International Publishing, pp. 188–197. ISBN: 978-3-319-08864-8. DOI: 10.1007/978-3-319-08864-8_18. URL: http://dx.doi.org/10.1007/978-3-319-08864-8_18.

– (July 2015). "Closed-loop acquisition of behaviour on the Sphero robot". In: *Proceedings of the European Conference on Artificial Life 2015*. Ed. by Paul Andrews et al. Complex Adaptive Systems. MIT Press, pp. 472–478. DOI: 10.7551/978-0-262-33027-5-ch084.

Berthold, Oswald, Mathias Müller, and Verena V. Hafner (2011). "A quadrotor platform for bio-inspired navigation experiments". In: *International workshop on bio-inspired robots*.

Bialek, W. and N. Tishby (Feb. 1999). "Predictive Information". In: *eprint arXiv:cond-mat/9902341*.

Bialek, William, Ilya Nemenman, and Naftali Tishby (2001). "Predictability, complexity, and learning". In: *Neural computation* 13.11, pp. 2409–2463.

Biswal, B. B. et al. (Mar. 2010). "Toward discovery science of human brain function". en. In: *Proceedings of the National Academy of Sciences* 107.10, pp. 4734–4739. ISSN: 0027-8424, 1091-6490. DOI: 10.1073/pnas.0911855107. URL: http://www.pnas.org/cgi/doi/10.1073/pnas.0911855107 (visited on 03/01/2018).

Braitenberg, Valentino (1984). *Vehicles - Experiments in Synthetic Psychology*. MIT Press.

Bray, Dennis (2009). *Wetware - A Computer in Every Living Cell*. Yale University Press.

Buhrmann, Thomas, Ezequiel Alejandro Di Paolo, and Xabier Barandiaran (2013). "A dynamical systems account of sensorimotor contingencies". In: *Frontiers in psychology* 4, p. 285.

Butko, Nicholas J. and Jochen Triesch (2007). "Learning sensory representations with intrinsic plasticity". In: 1.

Charnov, Eric L. (Apr. 1976). "Optimal foraging, the marginal value theorem". en. In: *Theoretical Population Biology* 9.2, pp. 129–136. ISSN: 00405809. DOI: 10.1016/0040-5809(76)90040-X. URL: http://linkinghub.elsevier.com/retrieve/pii/004058097690040X (visited on 05/18/2018).

Clark, Andy (2015). "Embodied Prediction". In: *Open MIND*. Ed. by Thomas K. Metzinger and Jennifer M. Windt. Frankfurt am Main: MIND Group. ISBN: 978-3-95857-011-5. DOI: 10.15502/9783958570115. URL: http://open-mind.net/papers/embodied-prediction.

Clune, J., J.-B. Mouret, and H. Lipson (Jan. 2013). "The evolutionary origins of modularity". en. In: *Proceedings of the Royal Society B: Biological Sciences* 280.1755, pp. 20122863–20122863. ISSN: 0962-8452, 1471-2954. DOI: 10.1098/rspb.2012.2863. URL: http:

//rspb.royalsocietypublishing.org/cgi/doi/10.1098/rspb.2012.2863 (visited on 06/09/2018).

Cook, Matthew and Jehoshua Bruck (2004). "Networks of Relations for Representation, Learning, and Generalization". In: *Proceedings of the Fourth International Conference on Intelligent System Design and Applications*. Ed. by Ajith Abraham and Janos Abonyi. Advances in Soft Computing. Springer-Verlag.

Cook, Matthew, Florian Jug, et al. (2010). "Unsupervised Learning of Relations." In: *ICANN (1)*. Ed. by Konstantinos I. Diamantaras, Wlodek Duch, and Lazaros S. Iliadis. Vol. 6352. Lecture Notes in Computer Science. Springer, pp. 164–173. ISBN: 978-3-642-15818-6. URL: http://dblp.uni-trier.de/db/conf/icann/icann2010-1.html#CookJKS10.

Copete, Jorge L., Yukie Nagai, and Minoru Asada (2016). "Motor development facilitates the prediction of others' actions through sensorimotor predictive learning". In: *Proceedings of the 6th IEEE International Conference on Development and Learning and on Epigenetic Robotics*.

Cover, Thomas M. and Joy A. Thomas (2006). *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience. ISBN: 0-471-24195-4.

Craik, Kenneth J. W. (1943). *The Nature of Explanation*. Cambridge University Press.

Crutchfield, James P. (1990). "Information and Its Metric". In: *Nonlinear Structures in Physical Systems*. Ed. by Lui Lam and Hedley C. Morris. New York, NY: Springer New York, pp. 119–130. ISBN: 978-1-4612-3440-1.

Crutchfield, James P. and David P. Feldman (2003). "Regularities unseen, randomness observed: Levels of entropy convergence". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 13.1, pp. 25–54. DOI: 10.1063/1.1530990. URL: https://doi.org/10.1063/1.1530990.

Cully, Antoine et al. (2015). "Robots that can adapt like animals". In: *Nature* 521, pp. 503–507. DOI: 10.1038/nature14422.

Dayan, Peter (2002). "Matters Temporal". In: *Trends in Cognitive Sciences* 6.3, pp. 105–106.

Dayan, Peter and Yael Niv (2008). "Reinforcement learning: The Good, The Bad and The Ugly". In: *Current Opinion in Neurobiology* 18.2, pp. 185–196. ISSN: 0959-4388. DOI: http://dx.doi.org/10.1016/j.conb.2008.08.003. URL: http://www.sciencedirect.com/science/article/pii/S0959438808000767.

Demiris, Yiannis and Anthony Dearden (2005). "From motor babbling to hierarchical learning by imitation: a robot developmental pathway". In: 123. Ed. by Luc Berthouze et al. URL: http://cogprints.org/4961/.

Demiris, Yiannis and Bassam Khadhouri (2006). "Hierarchical attentive multiple models for execution and recognition of actions". In: *Robotics and Autonomous Systems* 54.5, pp. 361–369. ISSN: 0921-8890. DOI: http://dx.doi.org/10.1016/j.robot.2006.02.003. URL: //www.sciencedirect.com/science/article/pii/S0921889006000169.

Der, Ralf and Georg Martius (2012). *The Playful Machine: Theoretical Foundation and Practical Realization of Self-Organizing Robots*. Cognitive Systems Monographs. Springer Berlin Heidelberg. ISBN: 978-3-642-20253-7. URL: http://www.springer.com/us/book/9783642202520.

Der, Ralf, Ulrich Steinmetz, and Frank Pasemann (1999). "Homeokinesis - A new principle to back up evolution with learning". In: *Proc. Intl. Conf. on Computational Intelligence for Modelling, Control and Automation (CIMCA 99)*. Vol. 55. Concurrent Systems Engineering Series. Amsterdam: IOS Press, pp. 43–47. URL: citeseer.ist.psu.edu/der99homeokinesis.html.

Doncieux, Stephane et al. (2015). "Evolutionary Robotics: What, Why, and Where to". In: *Frontiers in Robotics and AI* 2.4. ISSN: 2296-9144. DOI: 10.3389/frobt.2015.00004. URL: http://www.frontiersin.org/evolutionary_robotics/10.3389/frobt.2015.00004/abstract.

Dugatkin, Lee Alan (2014). *Principles of animal behavior*. English. W. W. NORTON & COMPANY.

Fefferman, Charles, Sanjoy Mitter, and Hariharan Narayanan (Oct. 2013). "Testing the Manifold Hypothesis". In: *arXiv:1310.0425 [math, stat]*. arXiv: 1310.0425. URL: http://arxiv.org/abs/1310.0425 (visited on 06/09/2018).

Floreano, Dario and Claudio Mattiussi (2008). *Bio-Inspired Artificial Intelligence*. MIT Press.

Franz, Matthias O. and Hanspeter A. Mallot (2000). "Biomimetic robot navigation". In: *Robotics and Autonomous Systems* 30.1–2, pp. 133–153. ISSN: 0921-8890. DOI: 10.1016/S0921-8890(99)00069-X. URL: http://www.sciencedirect.com/science/article/pii/S092188909900069X.

Friston, Karl J. and Klaas E. Stephan (2007). "Free-energy and the brain". In: *Synthese* 159.3, pp. 417–458. ISSN: 1573-0964. DOI: 10.1007/s11229-007-9237-y. URL: http://dx.doi.org/10.1007/s11229-007-9237-y.

Fritzke, B. (1995). "A Growing Neural Gas Network Learns Topologies". In: *Advances in Neural Information Processing Systems 7*. MIT Press, pp. 625–632.

Gallagher, Shaun (Jan. 2000). "Philosophical conceptions of the self: implications for cognitive science". en. In: *Trends in Cognitive Sciences* 4.1, pp. 14–21. ISSN: 13646613. DOI: 10.1016/S1364-6613(99)01417-5. URL: http://linkinghub.elsevier.com/retrieve/pii/S1364661399014175 (visited on 06/03/2018).

Gerken, Andreas, Oswald Berthold, and Verena Hafner (2017). "Behavioral diversity through homeokinesis in a compliant robot". In: *Proceedings of the 7th Joint IEEE International Conference on Development and Learning and on Epigenetic Robotics (ICDL-EpiRob)*. Lisbon, Portugal.

Gershenson, Carlos and Nelson Fernandez (2012). "Complexity and information: Measuring emergence, self-organization, and homeostasis at multiple scales". In: *Complexity* 18, pp. 29–44. DOI: 10.1002/cplx.21424. URL: http://arxiv.org/abs/1205.2026.

Gerstner, Wulfram et al. (1996). "A neuronal learning rule for sub-millisecond temporal coding". In: *Nature* 383.6595, pp. 76–78. DOI: 10.1038/383076a0. URL: http://dx.doi.org/10.1038/383076a0.

Ghazi-Zahedi, Keyan and Johannes Rauh (2015). "Quantifying Morphological Computation based on an Information Decomposition of the Sensorimotor Loop". In: *CoRR* abs/1503.05113. URL: http://arxiv.org/abs/1503.05113.

Grassberger, Peter (Sept. 1986). "Toward a quantitative theory of self-generated complexity". In: *International Journal of Theoretical Physics* 25.9, pp. 907–938. ISSN: 1572-9575. DOI: 10.1007/BF00668821. URL: https://doi.org/10.1007/BF00668821.

Grondman, Ivo et al. (2012). "A survey of actor-critic reinforcement learning: Standard and natural policy gradients". In: *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on* 42.6, pp. 1291–1307.

Hafner, Verena V. et al. (2010). "An autonomous flying robot for testing bio-inspired navigation strategies". In: *Proceedings for the joint conference of ISR 2010 (41st International Symposium on Robotics) and ROBOTIK 2010 (6th German Conference on Robotics)*. VDE Verlag.

Haruno, Masahiko, Daniel M. Wolpert, and Mitsuo M. Kawato (Oct. 2001). "MOSAIC Model for Sensorimotor Learning and Control". In: *Neural Comput.* 13.10, pp. 2201–2220. ISSN: 0899-7667. DOI: 10.1162/089976601750541778. URL: http://dx.doi.org/10.1162/089976601750541778.

Haruno, Masahiko, Daniel M Wolpert, and Mitsuo Kawato (2003). "Hierarchical MOSAIC for movement generation". In: *International congress series*. Vol. 1250. Elsevier, pp. 575–590.

Hasselt, Hado van and M. A. Wiering (Apr. 2007). "Reinforcement Learning in Continuous Action Spaces". In: *Approximate Dynamic Programming and Reinforcement Learning, 2007. ADPRL 2007. IEEE International Symposium on*, pp. 272–279. DOI: 10.1109/ADPRL.2007.368199.

Hauser, Helmut et al. (2011). "Towards a theoretical foundation for morphological computation with compliant bodies". In: *Biological cybernetics* 105.5-6, pp. 355–370.

He, Kaiming et al. (2016). "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

Hebb, Donald O. (1949). *Organization of Behavior*. Wiley.

Hesslow, Germund (2012). "The current status of the simulation theory of cognition". In: *Brain Research* 1428, pp. 71–79. ISSN: 0006-8993. DOI: http://dx.doi.org/10.1016/j.brainres.2011.06.026. URL: http://www.sciencedirect.com/science/article/pii/S0006899311011309.

Hoerzer, G. M., R. Legenstein, and W. Maass (2012). "Emergence of Complex Computational Structures From Chaotic Neural Networks Through Reward-Modulated Hebbian Learning". In: *Cerebral Cortex*. DOI: 10.1093/cercor/bhs348. URL: http://cercor.oxfordjournals.org/content/early/2012/11/09/cercor.bhs348.abstract.

Hogan, Neville and Dagmar Sternad (2012). "Dynamic primitives of motor behavior". English. In: *Biological Cybernetics* 106.11-12, pp. 727–739. ISSN: 0340-1200. DOI: 10.1007/s00422-012-0527-1. URL: http://dx.doi.org/10.1007/s00422-012-0527-1.

Horn, Berthold K. P. (1986). *Robot vision*. MIT electrical engineering and computer science series. MIT Press. ISBN: 978-0-262-08159-7.

Hwang, Jungsik et al. (2017). "Predictive Coding-based Deep Dynamic Neural Network for Visuomotor Learning". In: *arXiv preprint arXiv:1706.02444*.

Iida, Fumiya and Rolf Pfeifer (2004). ""Cheap" Rapid Locomotion of a Quadruped Robot: Self-Stabilization of Bounding Gait". In:

Iigaya, Kiyohito et al. (May 2017). "Learning Fast And Slow: Deviations From The Matching Law Can Reflect An Optimal Strategy Under Uncertainty". en. In: DOI: 10.1101/141309.

Ijspeert, Auke Jan (2008). "Central pattern generators for locomotion control in animals and robots: A review". In: *Neural Networks* 21.4, pp. 642–653.

Ijspeert, Auke Jan, Jun Nakanishi, and Stefan Schaal (Jan. 2002). "Learning rhythmic movements by demonstration using nonlinear oscillators". In: vol. 1. Proceedings of the IEEE/RSJ Int. Conference on Intelligent Robots and Systems.

Ioannou, P. and J. Sun (1996). *Robust Adaptive Control*. Prentice Hall, Inc. URL: http://www-rcf.usc.edu/%20ioannou/Robust_Adaptive_Control.htm.

Johnson-Laird, P. N. (2004). "The history of mental models". In: *Psychology of reasoning: Theoretical and historical perspectives*. Psychology Press, p. 179.

Jonschkowski, Rico and Oliver Brock (2015). "Learning state representations with robotic priors". In: *Autonomous Robots* 39.3, pp. 407–428.

Jr, John H. Long (2016). "Modularity and Sparsity: Evolution of Neural Net Controllers in Physically Embodied Robots". In: 1.

Kaplan, Frédéric and Verena V. Hafner (2006). "Information-theoretic framework for unsupervised activity classification". In: *Advanced Robotics* 20.10, pp. 1087–1103. DOI: 10.1163/156855306778522514. URL: http://dx.doi.org/10.1163/156855306778522514.

Kaplan, Frédéric and Pierre-Yves Oudeyer (2004). "Maximizing learning progress: an internal reward system for development". In: *Embodied artificial intelligence*. Springer, pp. 259–270.

Klyubin, Alexander S., Daniel Polani, and Chrystopher L. Nehaniv (2008). "Keep Your Options Open: An Information-Based Driving Principle for Sensorimotor Systems". In: *PLOS ONE* 3.12, pp. 1–14. DOI: 10.1371/journal.pone.0004018. URL: http://dx.doi.org/10.1371%2Fjournal.pone.0004018.

Kober, Jens, B. Mohler, and Jan Peters (2008). "Learning perceptual coupling for motor primitives". In: *Proc. IROS*, pp. 834–839.

Kober, Jens and Jan Peters (2011). "Policy search for motor primitives in robotics". English. In: *Machine Learning* 84.1-2, pp. 171–203. ISSN: 0885-6125. DOI: 10.1007/s10994-010-5223-6. URL: http://dx.doi.org/10.1007/s10994-010-5223-6.

Koller, Daphne and Nir Friedman (2009). *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press. ISBN: 978-0-262-01319-2.

Kolodziejski, Christoph et al. (2008). "On the asymptotic equivalence between differential Hebbian and temporal difference learning using a local third factor". In: *Advances in Neural Information Processing Systems 21*. Ed. by D. Koller et al. Curran Associates, Inc., pp. 857–864. URL: http://papers.nips.cc/paper/3419-on-the-asymptotic-equivalence-between-differential-hebbian-and-temporal-difference-learning-using-a-local-third-factor.pdf.

Konczak, Jürgen (2005). "On the notion of motor primitives in humans and robots". In: 123. Ed. by Luc Berthouze et al. URL: http://cogprints.org/4963/.

Konda, Vijay R and John N Tsitsiklis (2000). "Actor-critic algorithms". In: *Advances in neural information processing systems*, pp. 1008–1014.

Lakoff, George and Rafael Nunez (2000). *Where mathematics comes from*. New York: Basic books.

Lang, Kevin J., Alex H. Waibel, and Geoffrey E. Hinton (Jan. 1990). "A Time-delay Neural Network Architecture for Isolated Word Recognition". In: *Neural Netw.* 3.1, pp. 23–43. ISSN: 0893-6080. DOI: 10.1016/0893-6080(90)90044-L. URL: http://dx.doi.org/10.1016/0893-6080(90)90044-L.

Legenstein, R. et al. (2010). "A Reward-Modulated Hebbian Learning Rule Can Explain Experimentally Observed Network Reorganization in a Brain Control Task". In: *The Journal of Neuroscience* 30.25, pp. 8400–8410. DOI: 10.1523/JNEUROSCI.4284-09.2010. URL: http://www.jneurosci.org/content/30/25/8400.abstract.

Lehman, Joel and Kenneth O Stanley (2008). "Exploiting open-endedness to solve problems through the search for novelty." In: *ALIFE*, pp. 329–336.

– (2011). "Abandoning objectives: Evolution through the search for novelty alone". In: *Evolutionary computation* 19.2, pp. 189–223.

Letzkus, Johannes J, Björn M Kampa, and Greg J Stuart (2006). "Learning rules for spike timing-dependent plasticity depend on dendritic synapse location." In: *J. Neurosci.* 26.41, pp. 10420–9. DOI: 10.1523/JNEUROSCI.2650-06.2006.

Lewis, Frank (Aug. 2009). "Reinforcement Learning and Adaptive Dynamic Programming for Feedback Control". In: 1.

Li, Chunyuan et al. (Apr. 2018). "Measuring the Intrinsic Dimension of Objective Landscapes". In: *arXiv:1804.08838 [cs, stat]*. arXiv: 1804.08838. URL: http://arxiv.org/abs/1804.08838 (visited on 04/30/2018).

Lizier, Joseph T. (2014). "JIDT: An information-theoretic toolkit for studying the dynamics of complex systems". In: *CoRR* abs/1408.3270. URL: http://arxiv.org/abs/1408.3270.

Lizier, Joseph T., Mikhail Prokopenko, and Albert Y. Zomaya (2014). "A Framework for the Local Information Dynamics of Distributed Computation in Complex Systems". In: *Guided Self-Organization: Inception*. Ed. by Mikhail Prokopenko. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 115–158. ISBN: 978-3-642-53734-9. DOI: 10.1007/978-3-642-53734-9_5. URL: http://dx.doi.org/10.1007/978-3-642-53734-9_5.

Loeb, Gerald E. (Dec. 2012). "Optimal isn't good enough". In: *Biological Cybernetics* 106.11, pp. 757–765. ISSN: 1432-0770. DOI: 10.1007/s00422-012-0514-6. URL: https://doi.org/10.1007/s00422-012-0514-6.

Lorenz, Konrad (Aug. 1973). *Die Rückseite des Spiegels*. Vol. 1. Piper Verlag, München.

Lotter, William, Gabriel Kreiman, and David Cox (2015). "Unsupervised Learning of Visual Structure using Predictive Generative Networks". In: *CoRR* abs/1511.06380. URL: http://arxiv.org/abs/1511.06380.

Lukoševičius, Mantas and Herbert Jaeger (2009). "Reservoir computing approaches to recurrent neural network training". In: *Computer Science Review* 3.3, pp. 127–149. ISSN: 1574-0137. DOI: http://dx.doi.org/10.1016/j.cosrev.2009.03.005. URL: http://www.sciencedirect.com/science/article/pii/S1574013709000173.

Lungarella, Max, Teresa Pegors, et al. (Sept. 2005). "Methods for quantifying the informational structure of sensory and motor data". In: *Neuroinformatics* 3.3, pp. 243–262. ISSN: 1559-0089. DOI: 10.1385/NI:3:3:243. URL: https://doi.org/10.1385/NI:3:3:243.

Lungarella, Max and Olaf Sporns (2006). "Mapping Information Flow in Sensorimotor Networks". In: *PLOS Computational Biology* 2.10, pp. 1–12. DOI: 10.1371/journal.pcbi.0020144. URL: https://doi.org/10.1371/journal.pcbi.0020144.

Lungarella, M et al. (Dec. 2003). "Developmental robotics: a survey". In: *CONNECTION SCIENCE* 15.4, pp. 151–190. ISSN: 0954-0091. DOI: 10.1080/09540090310001655110.

Markram, Henry et al. (1997). "Regulation of Synaptic Efficacy by Coincidence of Postsynaptic APs and EPSPs". In: *Science* 275.5297, pp. 213–215. ISSN: 0036-8075. DOI: 10.1126/science.275.5297.213. URL: http://science.sciencemag.org/content/275/5297/213.

Martius, Georg, Ralf Der, and Nihat Ay (2013). "Information driven self-organization of complex robotic behaviors". In: *PloS one* 8.5, e63400. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0063400. URL: http://europepmc.org/articles/PMC3664628.

Martius, Georg, Luisa Jahn, et al. (2014). "Self-exploration of the Stumpy Robot with Predictive Information Maximization". English. In: *From Animals to Animats 13*. Ed. by AngelP. del Pobil et al. Vol. 8575. Springer International Publishing, pp. 32–42. ISBN: 978-3-319-08863-1. DOI: 10.1007/978-3-319-08864-8_4. URL: http://dx.doi.org/10.1007/978-3-319-08864-8_4.

Martius, Georg and Eckehard Olbrich (July 2015). *Quantifying Self-Organizing Behavior of Autonomous Robots*. Vol. 1.

Mattingly, Henry H. et al. (Feb. 2018). "Maximizing the information learned from finite data selects a simple model". In: *Proceedings of the National Academy of Sciences* 115.8. arXiv: 1705.01166, pp. 1760–1765. ISSN: 0027-8424, 1091-6490. DOI: 10.1073/pnas.1715306115. URL: http://arxiv.org/abs/1705.01166 (visited on 04/19/2018).

Michalski, Vincent, Roland Memisevic, and Kishore Konda (2014). "Modeling Deep Temporal Dependencies with Recurrent Grammar Cells". In: *Advances in Neural Information Processing Systems 27*. Ed. by Z. Ghahramani et al. Curran Associates, Inc., pp. 1925–1933. URL: http://papers.nips.cc/paper/5549-modeling-deep-temporal-dependencies-with-recurrent-grammar-cells.pdf.

Mischiati, Matteo et al. (2015). "Internal models direct dragonfly interception steering". In: *Nature* 517.7534, p. 333.

Monod, Jacques (1972). *Chance and Necessity*. Collins.

Montroll, Elliot W (1956). "Random walks in multidimensional spaces, especially on periodic lattices". In: *Journal of the Society for Industrial and Applied Mathematics* 4.4, pp. 241–260.

Morse, A. F. et al. (Dec. 2010). "Epigenetic Robotics Architecture (ERA)". In: *IEEE Transactions on Autonomous Mental Development* 2.4, pp. 325–339. ISSN: 1943-0604. DOI: 10.1109/TAMD.2010.2087020.

Moulin-Frier, Clément, Pierre Rouanet, and Pierre-Yves Oudeyer (Oct. 2014). "Explauto: an open-source Python library to study autonomous exploration in developmental robotics". In: *ICDL-Epirob - International Conference on Development and Learning, Epirob*. Genoa, Italy. URL: https://hal.inria.fr/hal-01061708.

Müller, Vincent C and Matej Hoffmann (2017). "What is morphological computation? On how the body contributes to cognition and control". In: *Artificial life* 23.1, pp. 1–24.

Mussa-Ivaldi, F. A. and S. A. Solla (2004). "Neural Primitives for Motion Control". In: *IEEE Journal of Oceanic Engineering* 29, p. 640.

Nakajima, Kohei, Nico M. Schmidt, and Rolf Pfeifer (2014). "Measuring information transfer in a soft robotic arm". In: *CoRR* abs/1407.4162. URL: http://arxiv.org/abs/1407.4162.

Nardi, Renzo De et al. (July 2006). "Evolution of Neural Networks for Helicopter Control: Why Modularity Matters". In: *Proceedings of the IEEE Congress on Evolutionary Computation*.

Ng, Andrew Y. and H. Jin Kim (Jan. 2004). "Stable adaptive control with online learning". In: vol. 1. In Proc. NIPS. MIT Press.

Nielsen, Frank and Richard Nock (Jan. 2014). "On the Chi square and higher-order Chi distances for approximating f-divergences". In: *IEEE Signal Processing Letters* 21.1. arXiv: 1309.3029,

pp. 10–13. ISSN: 1070-9908, 1558-2361. DOI: 10.1109/LSP.2013.2288355. URL: http://arxiv.org/abs/1309.3029 (visited on 05/27/2018).

Niv, Yael (2009). "Reinforcement learning in the brain". In: *Journal of Mathematical Psychology* 53.3, pp. 139–154. ISSN: 0022-2496. DOI: http://dx.doi.org/10.1016/j.jmp.2008.12.005. URL: http://www.sciencedirect.com/science/article/pii/S0022249608001181.

O'Regan, J Kevin and Alva Noë (2001). "A sensorimotor account of vision and visual consciousness". In: *Behavioral and brain sciences* 24.5, pp. 939–973.

Oudeyer, Pierre-Yves, Frédéric Kaplan, and Verena V. Hafner (Apr. 2007). "Intrinsic Motivation Systems for Autonomous Mental Development". In: *Evolutionary Computation, IEEE Transactions on* 11.2, pp. 265–286. ISSN: 1089-778X. DOI: 10.1109/TEVC.2006.890271.

Patraucean, Viorica, Ankur Handa, and Roberto Cipolla (2015). "Spatio-temporal video autoencoder with differentiable memory". In: *CoRR* abs/1511.06309. URL: http://arxiv.org/abs/1511.06309.

Pérez-Uribe, Andrés (2001). "Using a Time-Delay Actor-Critic Neural Architecture with Dopamine-Like Reinforcement Signal for Learning in Autonomous Robots". English. In: *Emergent Neural Computational Architectures Based on Neuroscience*. Ed. by Stefan Wermter, Jim Austin, and David Willshaw. Vol. 2036. Springer Berlin Heidelberg, pp. 522–533. ISBN: 978-3-540-42363-8. DOI: 10.1007/3-540-44597-8_37. URL: http://dx.doi.org/10.1007/3-540-44597-8_37.

Peters, J. and S. Schaal (2006). "Policy Gradient Methods for Robotics". In: *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pp. 2219–2225. DOI: 10.1109/IROS.2006.282564.

Pfeifer, Rolf and Josh Bongard (2007). *How the Body Shapes the Way We Think*. MIT Press.

Pfeifer, Rolf, Max Lungarella, and Fumiya Iida (2007). "Self-organization, embodiment, and biologically inspired robotics". In: *science* 318.5853, pp. 1088–1093.

Pfeifer, Rolf, Max Lungarella, Olaf Sporns, et al. (2007). "On the Information Theoretic Implications of Embodiment – Principles and Methods". In: *50 Years of Artificial Intelligence: Essays Dedicated to the 50th Anniversary of Artificial Intelligence*. Ed. by Max Lungarella et al. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 76–86. ISBN: 978-3-540-77296-5. DOI: 10.1007/978-3-540-77296-5_8. URL: https://doi.org/10.1007/978-3-540-77296-5_8.

Philipona, D., J. K. O'Regan, and J. P. Nadal (2003). "Is there something out there?: Inferring space from sensorimotor dependencies". In: *Neural computation* 15.9, pp. 2029–2049.

Poincaré, Henri (1905). *Science and Hypothesis*. THE WALTER SCOTT PUBLISHING CO., LTD.

Polani, Daniel, Olaf Sporns, and Max Lungarella (2007). "How Information and Embodiment Shape Intelligent Information Processing". In: *50 Years of Artificial Intelligence: Essays Dedicated to the 50th Anniversary of Artificial Intelligence*. Ed. by Max Lungarella et al. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 99–111. ISBN: 978-3-540-77296-5. DOI: 10.1007/978-3-540-77296-5_10. URL: https://doi.org/10.1007/978-3-540-77296-5_10.

Pólya, G. (1921). "Über eine Aufgabe der Wahrscheinlichkeitsrechnung betreffend die Irrfahrt im Straßennetz". und. In: *Mathematische Annalen* 84, pp. 149–160. ISSN: 0025-5831; 1432-1807/e. URL: https://eudml.org/doc/158886 (visited on 05/20/2018).

Porr, Bernd and Florentin Wörgötter (2003a). "Isotropic Sequence Order Learning". In: *Neural Computation* 15.4, pp. 831–864. URL: http://dblp.org/db/journals/neco/neco15.html#PorrW03.

– (2003b). "Isotropic-sequence-order learning in a closed-loop behavioural system". In: *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* 361.1811, pp. 2225–2244. DOI: 10.1098/rsta.2003.1273.

Punjani, Ali and Pieter Abbeel (2015). "Deep learning helicopter dynamics models". In: *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, pp. 3223–3230.

Rao, Rajesh P. N. and Terrence J. Sejnowski (Oct. 2001). "Spike-Timing-Dependent Hebbian Plasticity As Temporal Difference Learning". In: *Neural Comput.* 13.10, pp. 2221–2237. ISSN: 0899-7667. DOI: 10.1162/089976601750541787. URL: http://dx.doi.org/10.1162/089976601750541787.

Rashevsky, N. (1973). "Chapter 2B - The Principle of Adequate Design". In: *Foundations of Mathematical Biology*. Ed. by Robert Rosen. Academic Press, pp. 143–175. ISBN: 978-0-12-597203-1. DOI: 10.1016/B978-0-12-597203-1.50010-5. URL: http://www.sciencedirect.com/science/article/pii/B9780125972031500105.

Rescorla, R. A. and A. R. Wagner (1972). "A theory of Pavlovian conditioning: variations in the effectiveness of reinforcement and nonreinforcement". In: *Classical conditioning II: Current research and theory*. Ed. by A. H. Black and W. F. Prokasy. Appleton-Century-Crofts, New York, pp. 64–99.

Rolf, M. and M. Asada (Aug. 2015). "What are goals? And if so, how many?" In: *2015 Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*, pp. 332–339. DOI: 10.1109/DEVLRN.2015.7346167.

Rolf, Matthias, Jochen J. Steil, and Michael Gienger (Aug. 2011). "Online Goal Babbling for rapid bootstrapping of inverse models in high dimensions". In: *Development and Learning (ICDL), 2011 IEEE International Conference on*. Vol. 2, pp. 1–8. DOI: 10.1109/DEVLRN.2011.6037368.

Rosen, Robert (2012). *Anticipatory Systems - Philosophical, Mathematical, and Methodological Foundations*. 2nd ed. Vol. 1. IFSR International Series on Systems Science and Engineering. Springer-Verlag New York. DOI: 10.1007/978-1-4614-1269-4.

Rosenblueth, Arturo, Norbert Wiener, and Julian Bigelow (1943). "Behavior, purpose and teleology". In: *Philosophy of science* 10.1, pp. 18–24.

Rössler, Otto E (1981). "An artificial cognitive-plus-motivational system". In: *Progress in Theoretical Biology* 6.147-160, p. 10.

Rössler, Otto E. (1974). "Adequate locomotion strategies for an abstract organism in an abstract environment - A relational approahc to brain function". In: *Lecture Notes in Biomathematics - Physics and Mathematics of the Nervous System*. Ed. by M. Conrad, W. Guttinger, and M. Dal Cin. Springer.

Rubner, Yossi, Carlo Tomasi, and Leonidas J Guibas (2000). "The Earth Mover's Distance as a Metric for Image Retrieval". en. In: p. 23.

Rummery, G. A. and M. Niranjan (1994). *On-Line Q-Learning Using Connectionist Systems*. Tech. rep. TR 166. Cambridge, England: Cambridge University Engineering Department.

Russell, Stuart J. and Peter Norvig (2003). *Artificial Intelligence - A modern approach*. Prentice Hall.

Saigusa, Tetsu et al. (2008). "Amoebae anticipate periodic events". In: *Physical review letters* 100.1, p. 018101.

Salge, Christoph, Cornelius Glackin, and Daniel Polani (2013). "Empowerment - an Introduction". In: *CoRR* abs/1310.1863. URL: http://arxiv.org/abs/1310.1863.

Schaal, Stefan, Peyman Mohajerian, and Auke Ijspeert (2007). "Dynamics systems vs. optimal control - a unifying view". In: *Progress in Brain Research*. Vol. 165, pp. 425–445. DOI: 10.1016/S0079-6123(06)65027-9.

Schaal, Stefan, Jan Peters, et al. (2005). "Learning movement primitives". In: *Robotics research. the eleventh international symposium*. Springer, pp. 561–572.

Scheier, Christian, Rolf Pfeifer, and Yasuo Kunyioshi (1998). "Embedded neural networks: exploiting constraints". In: *Neural Networks* 11.7-8, pp. 1551–1569.

Schillaci, Guido and Verena V. Hafner (2011). "Random movement strategies in self-exploration for a humanoid robot". In: *Proceedings of the 6th international conference on Human-robot interaction*, pp. 245–246.

Schillaci, Guido, Verena V. Hafner, and Bruno Lara (2016). "Exploration Behaviors, Body Representations, and Simulation Processes for the Development of Cognition in Artificial Agents". English. In: *Frontiers in Robotics and AI* 3. ISSN: 2296-9144. DOI: 10.3389/frobt.2016.00039. URL: https://www.frontiersin.org/articles/10.3389/frobt.2016.00039/full (visited on 06/01/2018).

Schmidhuber, Juergen (June 2009). *Driven by Compression Progress: A Simple Principle Explains Essential Aspects of Subjective Beauty, Novelty, Surprise, Interestingness, Attention, Curiosity, Creativity, Art, Science, Music, Jokes*. Vol. 1.

Schmidhuber, Jürgen (1991a). "A possibility for implementing curiosity and boredom in model-building neural controllers". In: *Proc. of the international conference on simulation of adaptive behavior: From animals to animats*, pp. 222–227.

– (Nov. 1991b). "Curious model-building control systems". In: *Neural Networks, 1991. 1991 IEEE International Joint Conference on*, 1458–1463 vol.2. DOI: 10.1109/IJCNN.1991.170605.

Schreiber, Thomas (July 2000). "Measuring Information Transfer". In: *Phys. Rev. Lett.* 85.2, pp. 461–464. DOI: 10.1103/PhysRevLett.85.461. URL: https://link.aps.org/doi/10.1103/PhysRevLett.85.461.

Schultz, Wolfram, Peter Dayan, and P. Read Montague (1997). "A Neural Substrate of Prediction and Reward". In: *Science* 275.5306, pp. 1593–1599. ISSN: 0036-8075. DOI: 10.1126/science.275.5306.1593. URL: http://science.sciencemag.org/content/275/5306/1593.

Schunck, Brian G. and Berthold K. P. Horn (1981). "Determining Optical Flow". In: *Artificial Intelligence* 17, pp. 185–203.

Sejnowski, Terrence, Sumantra Chattarji, and Patric Stanton (1989). "Induction of Synaptic Plasticity by Hebbian Covariance in the Hippocampus". In: ed. by Richard Durbin, Christopher Miall, and Graeme Mitchison. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., pp. 105–124. ISBN: 0-201-18348-X. URL: http://dl.acm.org/citation.cfm?id=103938.103945.

Seth, Anil K. (Apr. 2014). "A predictive processing theory of sensorimotor contingencies: Explaining the puzzle of perceptual presence and its absence in synesthesia". In: *Cognitive Neu-*

*roscience* 5.2, pp. 97–118. ISSN: 1758-8928. DOI: 10.1080/17588928.2013.877880. URL: https://doi.org/10.1080/17588928.2013.877880 (visited on 06/11/2018).

Shannon, Claude E. (1948). "A Mathematical Theory of Communication". In: 1.

Steels, Luc (2004). "The autotelic principle". In: *Embodied artificial intelligence*. Springer, pp. 231–242.

Stefano Nolfi, Dario Floreano (2000). *Evolutionary Robotics*. MIT Press.

Suri, R. E. and W. Schultz (1999). "A Neural Network Model with Dopamine-Like Reinforcement Signal That Learns a Spatial Delayed Response Task." In: *Neuroscience* 91, pp. 871–890.

Sutton, Richard S. and Andrew G. Barto (1998). *Reinforcement learning: an introduction*. 1st. Cambridge, MA, USA: MIT Press. ISBN: 0-262-19398-1.

Tam, David (2007). "Theoretical Analysis of Cross-Correlation of Time-Series Signals Computed by a Time-Delayed Hebbian Associative Learning Neural Network". In: *The Open Cybernetics & Systemics Journal* 1.1, pp. 1–4. DOI: 10.2174/1874110X07010110X0.

Tedrake, Russ et al. (Aug. 2009). "Learning to Fly like a Bird". In: vol. 1.

Terekhov, Alexander V. and J. Kevin O'Regan (2016). "Space as an Invention of Active Agents". In: *Frontiers in Robotics and AI* 3, p. 4. ISSN: 2296-9144. DOI: 10.3389/frobt.2016.00004. URL: http://journal.frontiersin.org/article/10.3389/frobt.2016.00004.

Thrun, Sebastian, Wolfram Burgard, and Dieter Fox (2000). *Probabilistic Robotics*. Early Draft - Not for distribution.

Tin, Chung and Chi-Sang Poon (2005). "Internal models in sensorimotor integration: perspectives from adaptive control theory". In: *Journal of Neural Engineering* 2.3, S147.

Todorov, Emanuel and Zoubin Ghahramani (Jan. 2003). "Unsupervised Learning of Sensory-Motor Primitives". In: *Proceedings of the 25th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. Vol. 1. IEEE.

Tononi, G, O Sporns, and G M Edelman (1994). "A measure for brain complexity: relating functional segregation and integration in the nervous system". In: *Proceedings of the National Academy of Sciences* 91.11, pp. 5033–5037. URL: http://www.pnas.org/content/91/11/5033.abstract.

Trewavas, Anthony (Aug. 2014). *Plant Behaviour and Intelligence*. Vol. 1. OUP Oxford.

Van Rossum, Mark CW, Maria Shippi, and Adam B Barrett (2012). "Soft-bound synaptic plasticity increases storage capacity". In: *PLoS computational biology* 8.12, e1002836.

Verschure, Paul FMJ, Ben JA Kröse, and Rolf Pfeifer (1992). "Distributed adaptive control: The self-organization of structured behavior". In: *Robotics and Autonomous Systems* 9.3, pp. 181–196.

Wahlström, Niklas, Thomas B Schön, and Marc Peter Deisenroth (2015). "From pixels to torques: Policy learning with deep dynamical models". In: *arXiv preprint arXiv:1502.02251*.

Weisstein, Eric W. (2018). *Pólya's Random Walk Constants*. en. Text. URL: http://mathworld.wolfram.com/PolyasRandomWalkConstants.html (visited on 05/18/2018).

Weng, Juyang et al. (2001). "Autonomous Mental Development by Robots and Animals". In: *Science* 291.5504, pp. 599–600. ISSN: 0036-8075. DOI: 10.1126/science.291.5504.599. URL: http://science.sciencemag.org/content/291/5504/599.

Wibral, Michael, Nicolae Pampu, et al. (2013). "Measuring Information-Transfer Delays". In: *PLOS ONE* 8.2, pp. 1–19. DOI: 10.1371/journal.pone.0055809. URL: https://doi.org/10.1371/journal.pone.0055809.

Wibral, Michael, Viola Priesemann, et al. (2015). "Partial information decomposition as a unified approach to the specification of neural goal functions". In: *Brain and cognition*.

Wiener, Norbert (1949). *Extrapolation, Interpolation, and Smoothing of Stationary Time Series:With Engineering Applications*. MIT Press.

Williams, P. L. and R. D. Beer (Apr. 2010). "Nonnegative Decomposition of Multivariate Information". In: *ArXiv e-prints*.

Winfield, Alan F. T. and Verena V. Hafner (2018). "Anticipation in Robotics". In: *Handbook of Anticipation: Theoretical and Applied Aspects of the Use of Future in Decision Making*. Ed. by Roberto Poli. Cham: Springer International Publishing, pp. 1–30. ISBN: 978-3-319-31737-3. DOI: 10.1007/978-3-319-31737-3_73-1. URL: https://doi.org/10.1007/978-3-319-31737-3_73-1.

Wolpert, Daniel M, Zoubin Ghahramani, and Michael I Jordan (1995). "An internal model for sensorimotor integration". In: *Science* 269.5232, pp. 1880–1882.

Wolpert, Daniel M and Mitsuo Kawato (1998). "Multiple paired forward and inverse models for motor control". In: *Neural networks* 11.7-8, pp. 1317–1329.

Wörgötter, Florentin and Bernd Porr (2005). "Temporal Sequence Learning, Prediction, and Control: A Review of Different Models and Their Relation to Biological Mechanisms." In: *Neural Computation* 17.2, pp. 245–319. URL: http://dblp.uni-trier.de/db/journals/neco/neco17.html#WorgotterP05.

# Figures

# List of Algorithms

# Appendices

# A. Point mass system

The point mass system is a simplified model of quadrotor motion in free space. The model can represent many aspects of rigid body robot motion. The simple arm model of explauto (Moulin-Frier, Rouanet, and Oudeyer 2014) for example, and used in (Benureau 2015) can be mapped on the point mass system state space without modification. The point mass system is primarily defined by its degrees of freedom and the order of the state update equations. For the quadrotor, this could be position in three-dimensional space and a second order equation to model force control. Sensor and motor mappings are modelled by linear projection from the intrinsic state space. Multivariate gaussian noise and configurable nonlinear transfer functions can be applied in a point wise manner on the transformation results. The primary motor to sensor delay can be configured by the *lag* parameter.
The point mass equations system is given by

$$
\begin{array}{rrcl}
\text{mass} & m_{\mathrm{pm}} & \in & \mathbb{R} \\
\text{DoF} & d_{\mathrm{pm}} & \in & \mathbb{N} \\
\text{order} & o_{\mathrm{pm}} & \in & \mathbb{N} \\
\text{lag} & \tau & \in & \mathbb{N} \\
\text{state} & \boldsymbol{s}_t & \in & \mathbb{R}^{d_{\mathrm{pm}} \cdot o} \\
 & \boldsymbol{s}_t & = & (\boldsymbol{s}_t^{\mathsf{pos}}, \boldsymbol{s}_t^{\mathsf{vel}}, \boldsymbol{s}_t^{\mathsf{acc}})^T \\
\text{motors} & \boldsymbol{u}_t & \in & \mathbb{R}^{d_{\mathsf{motor}}} \\
\text{noise} & \boldsymbol{n}^{\mathsf{acc|vel}} & \sim & \mathcal{N}(\boldsymbol{\mu}, \Sigma^s) \\
\text{input coupling matrix} & \boldsymbol{C} & = & \boldsymbol{I} + \mathcal{N}(0, \Sigma^{\boldsymbol{C}}) \\
\text{transfer function} & h(\cdot) & = & h_i \\
\text{Second order equation} & & & \\
 & \boldsymbol{s}_t^{\mathsf{acc}} & = & h(\boldsymbol{C} \cdot \boldsymbol{u}_{t-\tau}^{\mathsf{acc}}) + \boldsymbol{n}^{\mathsf{acc}} \\
 & \boldsymbol{s}_{t+1}^{\mathsf{vel}} & = & \boldsymbol{s}_t^{\mathsf{vel}} \cdot \boldsymbol{c}_f + (\boldsymbol{I} \cdot \boldsymbol{s}_t^{\mathsf{acc}}) \cdot dt + \boldsymbol{n}^{\mathsf{vel}} \\
 & \boldsymbol{s}_{t+1}^{\mathsf{pos}} & = & \boldsymbol{s}_t^{\mathsf{pos}} + \boldsymbol{s}_{t+1}^{\mathsf{vel}} \cdot dt
\end{array}
$$

The sensorimotor manifold of this systems can be exhaustively sampled, which is shown as a scatter plot and timeseries along the computation graph if this experiment in Figure A.1.

## A. Point mass system

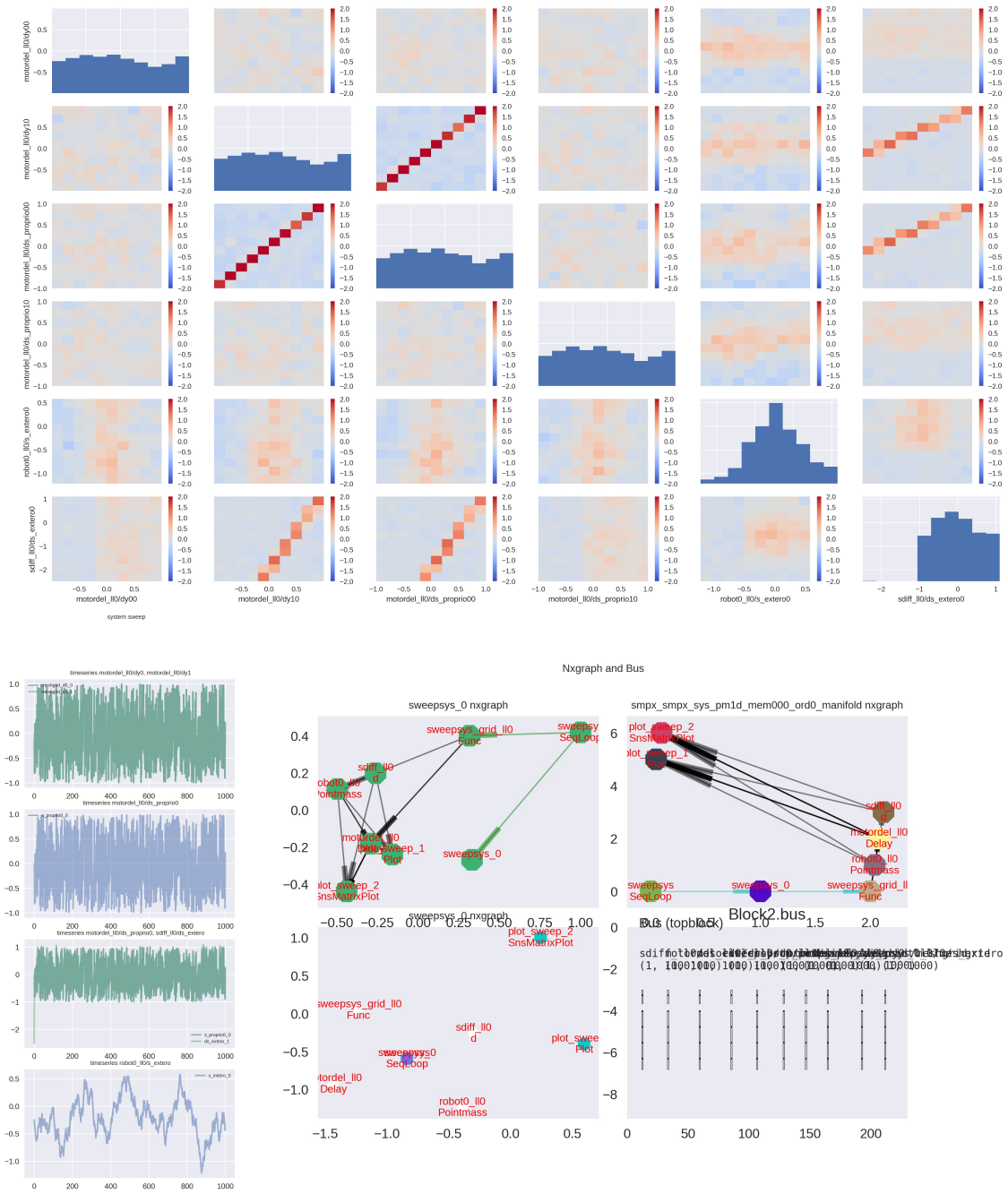



Figure A.1.: The scatter matrix of the 1D kinematic pointmass system is shown in the top. In the second row, the left column is the state timeseries, and the right panel is a visualization of the experiment graph.

# B. Low-level models

Detailed description of low-level learning algorithms used and implemented in the thesis need to be referred to the smp_base library documentation (Berthold 2018a).

# C. Additional experiments

## C.1. Robot experiments

Videos of the experiments described in this thesis are available on `https://vimeo.com/user12977458`.

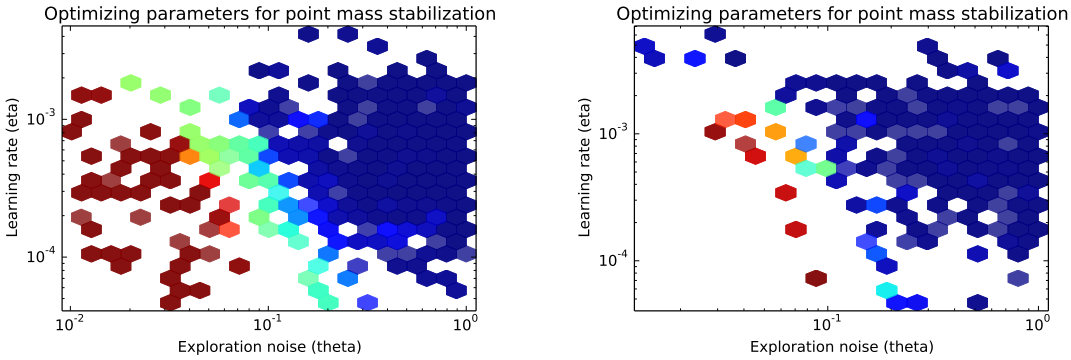
## C.2. Hyperparameter optimization statistics

The results of a hyperparameter optimization run using hyperopt (J. Bergstra, Yamins, and Cox 2013) with reservoir based Hebbian gradient search on a point mass goal tracking task are shown. The optimization occurred over the approximately complementary parameters learning rate $\eta$ and exploration noise $\theta$. Large exploration noise amplitude increases the uncertainty of prediction. The learning rate is proportional to the a priori confidence in the data, that is, the inverse of uncertainty. Thus, a partially linear relationship is predicted and shown in the parameter scan results in Figure C.1.

## C.3. Tappings and eligibility traces

### C.3.1. Introduction

For solving closed-loop motor learning tasks in continuous state-action spaces, three factor differential Hebbian learning rules can be applied to Single Hidden Layer networks. In addition to the pre- and postsynaptic terms, the third factor is *modulatory* and encodes a reward signal, derived from the task definition. Usually, the learning rules (LR) assume that if two signals are correlated they are correlated with unit delay. In arbitrary embodied systems whose control circuits run at a given sampling rate, the delay between an action and the return of the sensory consequence, on which the reward signal is based, is not necessarily one time-step but can be several and multiple time-steps. While in many cases it is possible to approximate the motor-sensor delay empirically, we would like to use a learning rule that requires less knowledge of the delay. An approach is presented, that applies the Hebbian update over an eligibility window of size $H$ of past states for the reward at each time step. For a system governed by a single time constant, the position in the window corresponding to the true delay will exclusively receive consistent reinforcement while for all other positions, the accumulated updates average to zero. Determining the motor delay then consists simply in finding the index of the maximum of the absolute values of accumulated weight updates over the eligibility window.




(a) Moderate noise

(b) Periodic perturbation

Figure C.1.: Performance landscape plotted over exploration noise amplitude and learning rate axes for two different environments. Performance $P$ is the negative squared error, lower being worse and shown in blue, higher values drawn in red indicating better ones. It can be seen on the right hand plot with more severe perturbations the set of viable solutions is significantly reduced with respect to learning parameters encoded in parameters $i$ of $\mathrm{A}_i$.

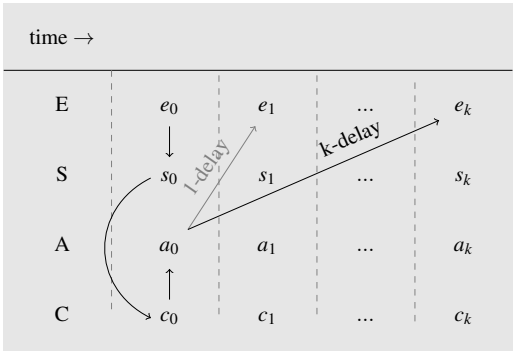
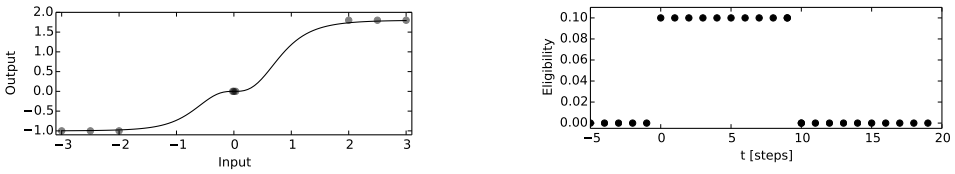



Figure C.2.: Temporal view of the sensorimotor loop with $E, S, A, C$ corresponding to environment, sensors, actuators and controller and discrete time. The shaded arrow indicates the usual unit delay scenario and the black arrow indicates a $k$-delay. We have, as a simplification, lumped all delay sources into the delay between network output and environment.




Motor transfer function

Eligibility window

Figure C.3.: The left panel shows the motor transfer function (a), the right panel shows the rectangular eligibility window (b).

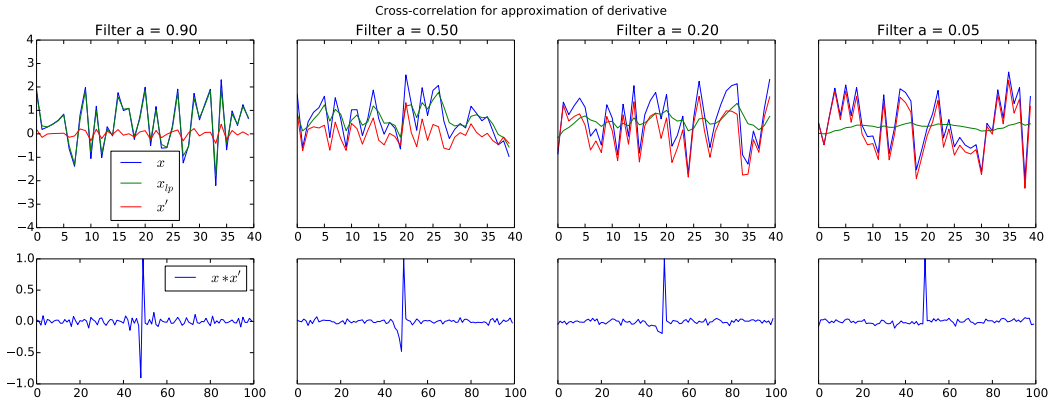



Figure C.4.: Cross-correlation functions of a noise signal $x$ and an approximated derivative $x'$ for different values of the recursive filter coefficient used in the approximation. In the top row we plot four different random signals $x$, along with a low-pass filtered $\bar{x}$ and the resulting point-wise difference $x'$. The bottom row displays the corresponding cross-correlation functions $(x \star x')[n]$. The true derivative has a clear antisymmetry around the correlation delay which obiously vanishes the closer $x'$ gets to $x$ when moving to right in the figure.

### Related work

Time-Delay Neural Networks (TDNN) have been devised for processing sequential data and can be used for computing cross-correlation functions (Lang, Waibel, and Hinton 1990), (Tam 2007). The difference to the current approach is, that time delay expansion is done in the learning stage at the output of the network.
A prediction based competitive actor-critic model targeting delayed response tasks has been proposed in (Suri and W. Schultz 1999) and applied to a robotic learning task in (Pérez-Uribe 2001). The correlation problem is a special case of the the Temporal Credit Assignment problem in Reinforcement Learning (RL) (Sutton and Andrew G. Barto 1998). Compared for example to the adaptive actor-critic architecture described in (A. G. Barto 1995), we consider the simpler case of immediate rewards with an unknown but fixed delay instead of the unconstrained reward prediction case. There is a line of inquiry that fully connects Temporal Difference (TD) methods from RL, three-factor differential Hebbian learning and Spike-Timing Dependent Plasticity (STDP) (Rao and T. J. Sejnowski 2001), (Porr and Wörgötter 2003a; Porr and Wörgötter 2003b), (Wörgötter and Porr 2005), (Kolodziejski et al. 2008). This provides a general framework within which the immediate reward with fixed delay is embedded.

## C.3.2. Model

### Network

Similar to previous approaches (Berthold and V. V. Hafner 2013a), a reservoir network, which is a special type of randomized recurrent neural network, is used in the current model as an input expansion. The expanded state is then combined as a weighted sum into the output signal. The output weights are learned with the Exploratory Hebbian (EH) learning rule (8.7) (Hoerzer,

Legenstein, and Maass 2012).

$$\Delta \boldsymbol{W}_{i,t}^{\text{out}} = \eta_{i,t} \boldsymbol{r}_{t-1} (y_{i,t-1} - \bar{y}_{i,t-1}) M_t \tag{C.1}$$

Here, $\eta \ll 1$ is a learning rate parameter, $\boldsymbol{r}$ is the reservoir state (the expanded input) and $M$ is a modulator. The expression within the parentheses serves as an approximation of the output derivative with $y$ beign the output and $\bar{y}$ a moving average of the output. The reservoir state evolves according to (C.3.2) with superscripts *res* and *in* of the matrices $\boldsymbol{W}$ denoting reservoir internal connectivity and input coupling, $\lambda$ a normalization factor scaling the spectral radius of $\boldsymbol{W}^{\text{res}}$ to a desired value, $\boldsymbol{u}$ being the network input, $\tau \in [0, 1]$ a time constant and $\nu_{\text{state}}$ the so-called state noise serving as a regularization.

$$\begin{aligned} \triangle \boldsymbol{x}_t &= \lambda \boldsymbol{W}^{\text{res}} \boldsymbol{r}_t + \boldsymbol{W}^{\text{in}} \boldsymbol{u}_t \\ \boldsymbol{x}_{t+1} &= (1 - \tau) \boldsymbol{x}_t + \tau \triangle \boldsymbol{x}_t \\ \boldsymbol{r}_{t+1} &= \tanh(\boldsymbol{x}_{t+1}) + \nu_{\text{state}} \end{aligned} \tag{C.2}$$

Finally, the network output is computed as

$$y_{i,t} = \boldsymbol{W}_{i,t}^{\text{out}} \boldsymbol{r}_t \tag{C.3}$$

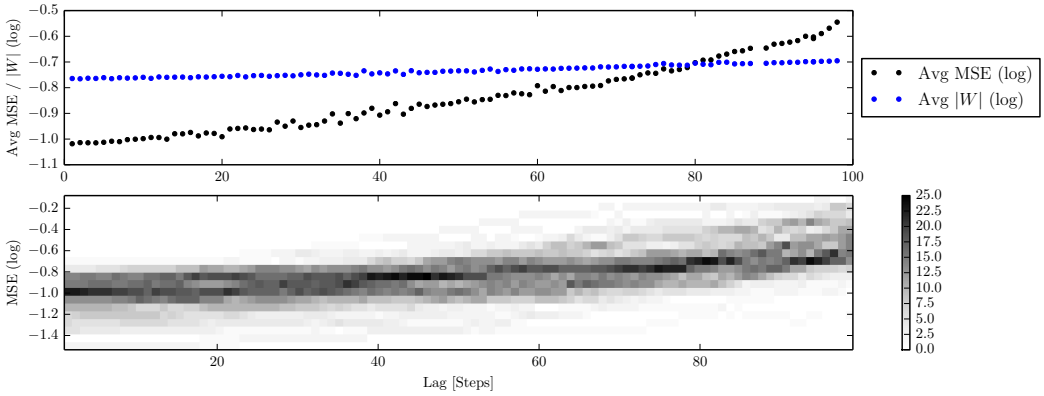



Figure C.5.: Top: Average MSE and output weight vector norm over motor delay parameter, bottom: histogram of MSEs over all runs.

### Robot

In this work the controlled system is simplified to a force controlled one-dimensional point mass with continuous disturbances of non-zero means. The motor transfer function has a deadband around zero and is asymmetric with respect to the input sign, see Figure C.3, left plot. In the open-loop case, the particle will just perform a biased random walk. In the closed-loop case, a controller is searched for, that stabilizes the particle at a given goal location, which constitutes the learning task.

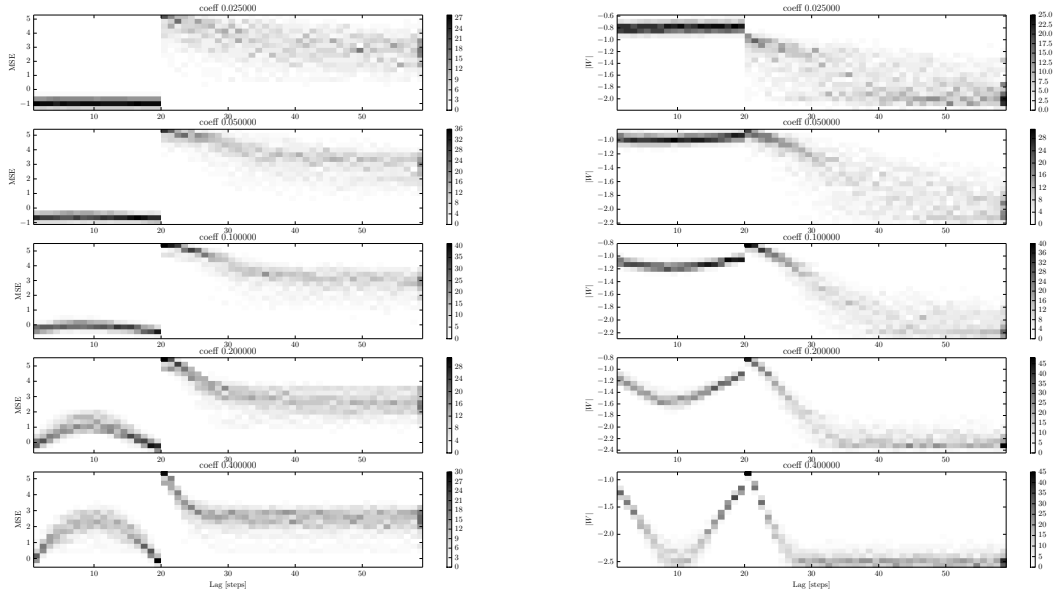



Figure C.6.: Motor delay sweep experiment covering the full eligibility window size and 40 steps beyond that. Five configurations with increasing filter coefficients are plotted with 50 runs per delay value. In the left column the logarithm of the final MSE in the tracking performance is plotted, in the right column the corresponding norm of the output weight vector. For every delay value, the system is allowed to learn the tracking task over 60000 time steps for 50 different randomized initial conditions (reservoir weights, disturbance parameters, ...) and collect the MSE for the last 1000 time-steps of each episode. For an increasing filter coefficient resulting in increasing responsivity of the filter it can be seen that the performance degrades, with a characteristic bulge for delay values lying in the center of the window. At the same time, a clear anti-correlation develops for delay values just beyond the window size. The distribution of the log MSE is bimodal which is caused by an asymmetry in the motor transfer function of the simulated robot particle. The two modes correspond to the two possible signs in direction along which the particle escapes.

### Temporal delay

For correlational learning rules in rate-based formulations it is necessary to know the sensorimotor delay. This is the time between the realization of a value at the network output and its return as a sensory consequence on the input of the network. The EH-rule as given in Equation 8.7 depends on the immediate response of the controlled system. To allow registering responses lying within an extended window following the occurence of one particular state we need to change the LR to accomodate an arbitrary motor delay $T$. This corresponds to the state becoming and remaining eligible for updates for some time after it has been visited. The extended rule is

$$\Delta \boldsymbol{W}^{\text{out}}_{i,t} = \eta_{i,t} \boldsymbol{r}_{t-T}(y_{i,t-T} - \bar{y}_{i,t-T}) M_t \quad \text{EH}(T)\text{-rule} \tag{C.4}$$

which can still accumulate the correlations using the fixed and known $T > 1$. A corresponding temporal sensorimotor loop diagram is given in Figure C.2. Getting rid of the dependence on the fixed and known delay $T$ for autonomous learning scenarios, it is necessary to estimate this delay, otherwise learning cannot pick up any correlations. In RL terms this is the Temporal Credit Assignment problem, although in a mild form. We introduce an eligibility trace as a replacement for a single time-step correlation and convolve the reward signal with all the state-action pairs lying within the elgibility window. The single time-step case can be regarded as a window of size one. The learning rule with eligibility traces is given by

$$\Delta \boldsymbol{W}^{\text{out}}_{i,t} = \textstyle\sum_k^{\text{H}} \eta_{i,t} h_k \boldsymbol{r}_{t-k}(y_{i,t-k} - \bar{y}_{i,t-k}) M_t \quad \text{EHE-rule} \tag{C.5}$$

with eligibility- or learning window $h$ and window size $H$. The simplest window function, which we also use here, is rectangular, see Figure C.3. This approach does not fully solve the motor delay problem but it relaxes it by having to specify an interval rather than a specific value.
The effect of the convolution depends on the properties of the cross-correlation function of the Hebbian and modulatory terms. In the approximate differential case ($\Delta y/\Delta t \approx y - \bar{y}$), the properties are determined by the coefficient of the low-pass filter used in constructing $\bar{y}$. A responsive filter leads to a pronounced anti-correlation for lag values neighbouring the real lag. Depending on the position of the actual lag within the eligibility window this can lead to a degradation of performance due to unwanted weight-contributions by anti-correlated terms, as illustrated in Figure C.4 for an artificial example signal.
As a side effect of using an eligibility trace, we can track which delay value within the window accumulates the largest weight contributions because on average, it will have the highest correlation with the reward signal. Thus, we can sketch an algorithm (Alg. 4) for concurrently learning and estimating the motor delay by observing the learning activity within the learning window.

## C.3.3. Results

We consider two questions: 1) what is the performance of the learner with a known singular motor delay under increasing the delay in the system, and 2) what is the behaviour of the learner when we are using a generalization of the single time-step learning window as proposed above. We learn a tracking task, that is, the inert particle has to be controlled with finite energy to stabilize position at a setpoint and follow abrupt setpoint changes. The performance measure is the Mean

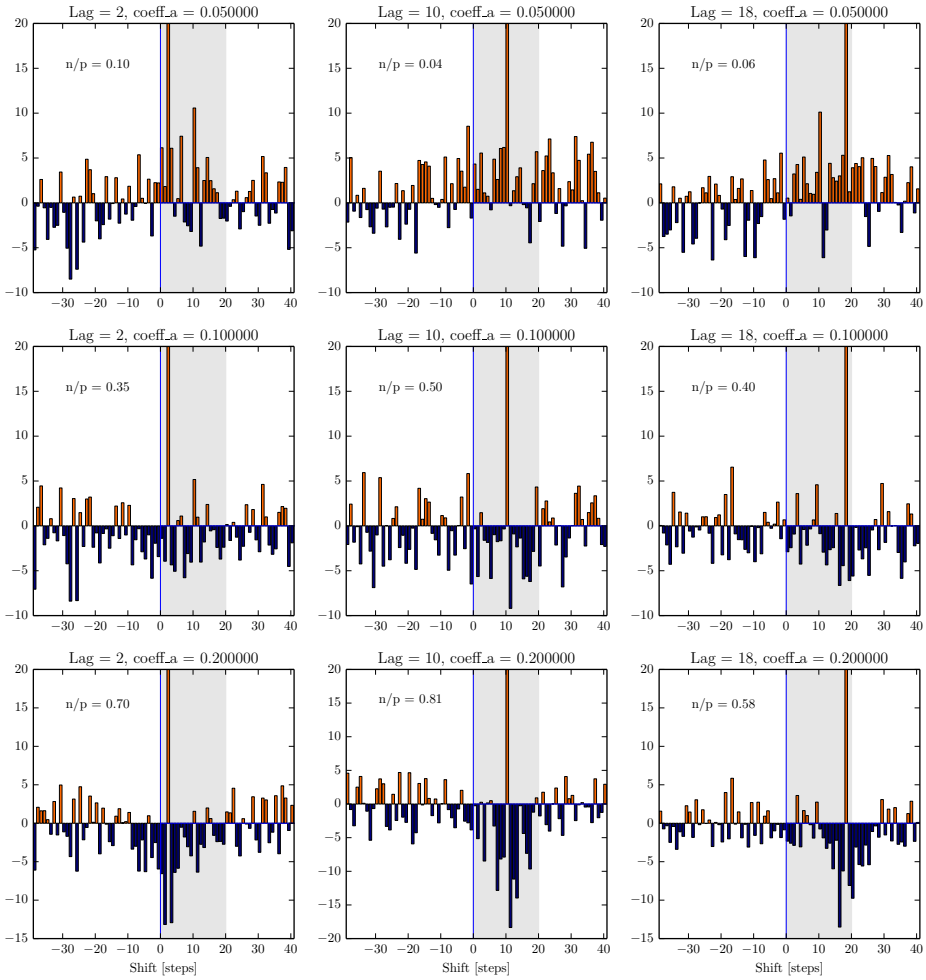



Figure C.7.: Empirical cross-correlation function of motor output $z$ and reward $p$ estimated over 10 runs per panel. Each row represents one filter coefficient and columns contain runs for one delay value. It can be seen how the shape of the correlation changes with the coefficient. Orange colors indicate positive correlation and blue negative correlation, $n/p$ is the quotient of the separate sums of positive and negative weight contributions. The closer this quotient is to zero, the more the true correlation contributed to the accumulated weights.

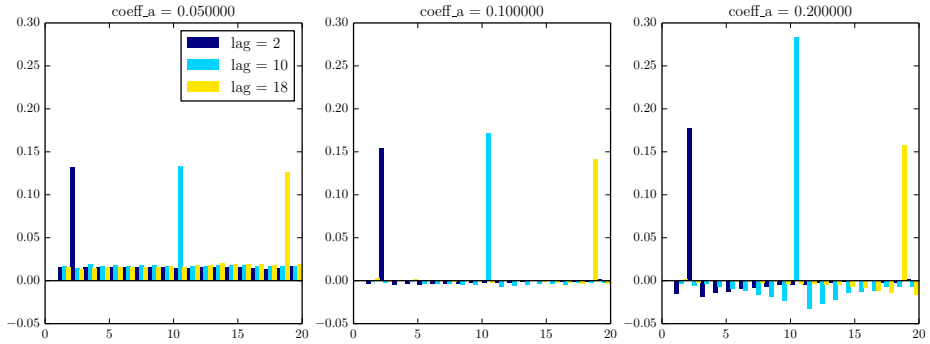



Figure C.8.: Accumulated weight contribution per eligbility window slot. This is the same as the cross-correlation function andcan be used to determine the motor delay by looking for maxima. Since this representation is used in the learning algorithm, the cross-correlation function does not need to be explicitly computed.

**Algorithm 4** EHE-rule with Find lag

$N$ = state dimensionality, $H$ = eligibility window size
cw $\leftarrow [0, ..., 0]$ ▷ Initialize contributed weights to zero for all lag values.
**repeat** ▷ forever
    exploration step $i$
    $\Delta w = \mathbf{0}$
    **for** $k$ in $[1, ..., H]$ **do**
        $\Delta w_k = \eta \cdot h_k \cdot r_{t-k} \cdot \frac{\Delta y}{\Delta t}_{t-k} \cdot M$ ▷ Apply learning rule
        $\Delta \text{cw}_k = \sum_{j=[1,...,N]} \Delta w_{k,j}$
        $\text{cw}_k = \text{cw}_k + \Delta \text{cw}_k$
        **if** $i >$ correlation convergence **then**
            lag $= \text{argmax}_k \, |\text{cw}_k|$
        **end if**
        $\Delta w = \Delta w + \Delta w_k$ ▷ Accumulate weight changes for learning step $i$
    **end for**
    $w = w + \Delta w$
**until** end of episode

Squared Error (MSE) of position and setpoint, summed over a fixed window at the end of the episode. As can be expected, the average MSE attained increases with the lag parameter. The results averaged over 1000 experiments are given in Figure C.5.

We now run the tracking task with an eligibility window of a fixed size for an unknown motor delay and vary the actual motor delay through a range covering three times the window size. We repeat the experiment for several different low-pass filter coefficients. The other hyperparameters of the learner are kept in the same ranges as opposed to the fixed lag single slot correlation experiment above. Only the learning rate had to be slightly lowered. The result is shown in Figure C.6. Depending only on the filter coefficient, the learning performance varies strongly.

Looking at the cross-correlation of two of the signals going into the learning rule, viz. motor output $z$ and reward $p$, see Figure C.7, helps to explain the observations. The region shaded in grey indicates that part of the cross-correlation function that lies within the eligibility window. The actual delay values always stick out clearly. In the plot they have been truncated for better readability of the values near the x-axis. Every bar in the plot represents a weighted contribution to the overall output weight vector for every possible delay value. For a coefficient of 0.05 there is only positive residual correlation. When the coefficient is increased, that is, the filter made more responsive, there appears an anti-correlated residual both left and right of the actual delay. Further increasing the coefficient makes this more pronounced. Now, if the real delay is near the window margins, the anti-correlated contributions are halfed as compared to the delay lying in the center of the window, resulting in an overall reduction of accumulated correlation. This explains the bulge in the plot of Figure C.6. In the plot we draw the quotient $n/p$ of negative to positive contributions to further illustrate the phenomenon.

In Figure C.8 we plot the accumulated weight contributions for three exemplary delay values (2, 10, 18 for a window size of 20) which basically provide the same picture. Interestingly, the changing shape of the cross-correlation function resembles the different STDP windows given in (Letzkus, Kampa, and Stuart 2006) for different dendritic distances, which in turn correspond to different effective delays between action potentials.

### C.3.4. Conclusion

Causal systems always exhibit a delay or *lag* between a cause and an effect. In an ideal system this can be one time step or sampling interval ($\Delta t$) but for real systems this can be anything larger than one time step. We also refer to this lag as the *motor delay*. For correlational learning the knowledge of the motor delay is essential for learning to work at all. When the delay is known, and for many systems it can be found, the knowledge can be embedded structurally into the design of the learner as a specific time-shift among the variables that need to be correlated. In other cases, when for some reason the lag cannot be determined or is variable we need to consider a separate mechanism for either determining the delay or modifying the learning rule so that it does not need precise knowledge about the lag.

We presented an extended differential Hebbian learning rule that performs a time delay expansion at the network output, constituting an elgibility trace. The learning rule was applied to a simulated motor learning task in which the actual motor delay information was withheld from the learner. Still, using this approach the learning system is able to acquire a stabilizing controller. We

performed a variation experiment for the parameter used in approximating the output derivative which turns out to have a strong effect on the success of learning.
The changing shape of the cross-correlation functions that appear in this experiment indicate an interesting relation to a similar type of learning on a smaller time scale, which suggests further inquiry. Furthermore, the anti-correlated component which can be read extracted from the system prior to the occurence of the corresponding reward can be interpreted as a reward prediction signal which could potentially be exploited for improving the learning performance.

## C.4. Learning internal models of quadrotor dynamics with open-loop exploration

### C.4.1. Intro

Learning of primitive motor skills on a quadrotor robot is investigated here. The approach is composed of three stages, open-loop self-exploration with increasing degrees of freedom, forward and inverse model-pair fitting and closed-loop model exploitation, where the performance of different learning machines on a position tracking task is compared.
Endowing robots with autonomous learning skills is an open research problem. In that context, learning of internal models for acquiring motor skills on robots has been shown to be an intuitive and effective approach for different systems (Weng et al. 2001; Lungarella et al. 2003; M. Asada et al. 2009). Here, learning a dynamics model of a force-controlled quadrotor is examined. A simple developmental schedule for the robot is devised, that enables it, to realize full four degree of freedom (4-DoF) control in a safe manner. Experiments are performed in simulation but are designed for transfer onto a real robot. The two-phase method of open-loop exploration of the sensorimotor space with subsequent fitting of a model in this data is also knownn as motor babbling. The schedule consists in initially restricting the degrees of freedom available to the robot, learning a model in the restricted space, and then explore the remaining space while already using the first model for control.

### C.4.2. Problem and related work

Consider a quadrotor with an attitude controller implanted. Given that system, it is necessary to generate attitude control inputs, that enable the quadrotor to hover on a spot. This is the same as requiring the quadrotor to hold the $x, y$ and $z$ position coordinates steady. The ability to stabilize a position also enables the robot to move to distant locations by moving the goal location. Ideally, the robot should be able to acquire this skill on its own by exploring its sensorimotor space, learning a model of its own dynamics in the environment and the exploit the learned model to perform different motion tasks. Motor babbling has been investigated on humanoids and robot arm systems (Demiris and Dearden 2005; Schillaci and V. V. Hafner 2011; Benureau, Fudal, and Oudeyer 2014). All of these are based on kinematic systems. The current aim is to evaluate open-loop exploration and learning for acquiring a /dynamic/ model of the robot. This is a challenging task when the robot is dynamically unstable and learning must be quick and safe.

The vertical component of the quadrotor motion is examined first. If the acceptable input range of the collective thrust of all four propellers is in the range from 0 to 1, the unperturbed ideal system will only hover at a system specific hover thrust, let's say 0.5. Anything above that value will make the quadrotor ascend and anything below will make it descend. Due to the integration of its control inputs performed by the system, velocity will grow quickly when thrust input is above nominal hover thrust, making it for example crash into the ceiling if the experiment is performed in a closed room. The same is true for the lateral motion components, x and y. It is desired to limit the volume of space used by the robot during exploration.

In the first stage of the experiment, the robot's $x, y$ position is fixed and only the vertical degree of freedom is explored. An approximate value for the nominal hover thrust is determined empirically and the exploration range is set to slightly exceed this value above and below, making it unlikely that the robot will move beyond a certain maximal altitude during open-loop exploration by random commands.

During the exploration episode, the sensorimotor data is recorded and stored in the matrix $\boldsymbol{S}$. The state space of this experiment is given by the six-dimensional ground truth pose. One model is fitted to each of the forward and inverse input configurations. The forward configuration means taking the sensory state at the current time $t$ and the motor command at $t$ and concatenating them to form the input to the model. The target in this configuration is the next sensory state at time $t + 1$. This model can act as a predictor of the next system state given a current state and a motor command.

In the inverse configuration, the current and the next state is concatenated into the model's input vector. The target is the current motor command. This configuration can be used to predict the motor command that will make the system move from the current state to the state at time $t+1$.

In the exploitation stage, several things can be done with the trained models. The inverse model can be used alone by just feeding it the current state as currently measured and a desired state, collecting the predicted motor command and feed that into the system. Problems arise, when the current and next state configurations have not been seen during training, because, for example, their mutual distance exceeds the system's maximum velocity. What happens in these cases depends on the learning machine used for learning the models.

Another thing that can be done is to query the forward model with the current state and different motor command candidates and selecting the one with the most desirable outcome. In addition, the forward model can be used to check the validity of the motor command suggested by the inverse model.

If this works as expected, the robot will display some kind of coherent behaviour while exploiting the internal models in this way. The sensorimotor data generated during exploitation can be used to train another separate model on this data. This data will be more detailed around the operating point of the stabilization behaviour generated by the initial model. The second model allows more precise control in that region of sensorimotor space after training, and this process could be repeated, driven by the average prediction error levels and other motivation mechanism.

These experiments have been conducted using the MORSE robot simulator with its supplied dynamics quadrotor model which provides all the configurabilty and control modes for our experiments. Using ROS, a realtime architecture the experiment is ready to be performed on a real robot flying within an optical tracking system.

## C.4.3. Experiments

### 1-DoF (vertical) motion

This simple experiment serves as an initial proof of feasibility of the methodology by restricting motion to one DoF. The quadrotor is operated with attitude control. At the beginning of an exploration episode, it is placed on a fixed origin on the floor inside a room with boundaries to each side and no ceiling for visibility. The attitude control mode requires a four-vector as input with the roll, pitch, yaw and thrust components. The first three are clamped to zero and the thrust value is just fed with uniform random values in the range 0.3 to 0.65. That range was determined experimentally to yield and overall trajectory of the z position that is bounded. The resulting position trajectory on the z-axis is shown in Figure C.9.
Create data:

```
# original version
python im_quadrotor_controller.py --deltatime 0.1 --len_episode 10000 -m motorbabbling_attitude_z
# better
python im_experiment.py --conf conf/im_quadrotor_motorbabbling_z.py
```

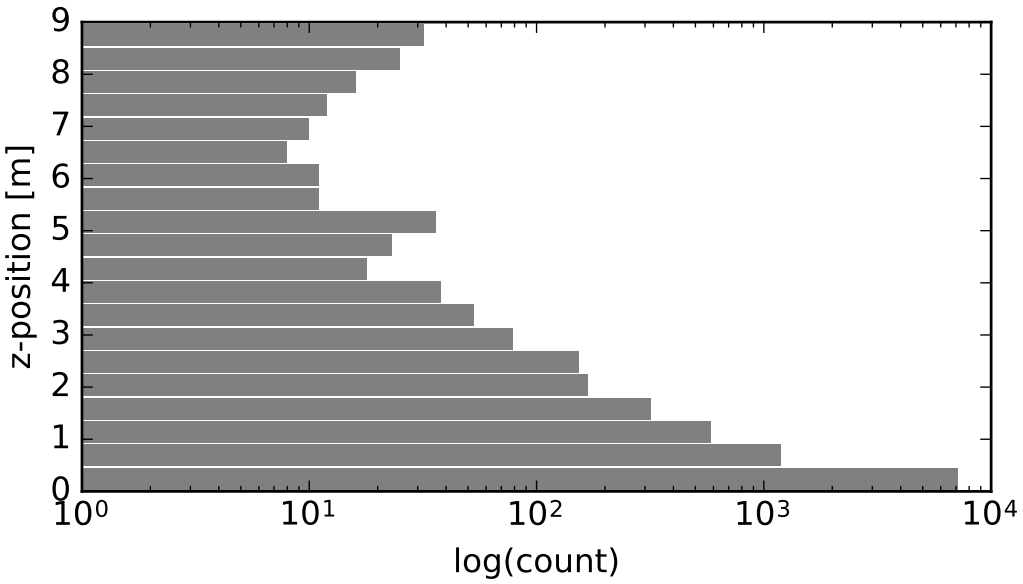



Figure C.9.: Histogram of z positions in an exploration episode of 10000 timesteps using random thrust values from the interval [0.3, 0.65].

Create plot:

```
python im_experiment.py --conf conf/im_quadrotor_plot_exploration_z_acc_thrust.py
```

The forward / inverse model pair training data is created by an embedding of the plain timeseries, specified as concatenation of time-shifted state vectors and motor commands into the respective training data matrix. Only a single time step temporal shift is considered, w.l.o.g. Evaluation shows a clear linear dependence of the vertical acceleration on thrust levels, and thus a linear model approximates the data sufficiently well in this case Table C.1.

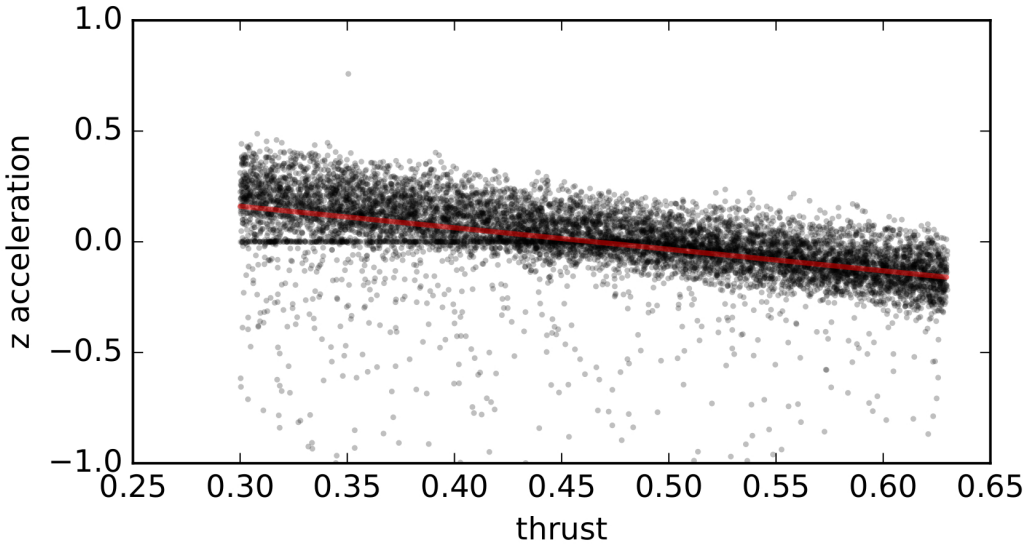



Figure C.10.: Plotting (z-axis) acceleration response over thrust reveals a clear linear relationship.

1. Training all models

```
python im_experiment.py --conf conf/im_quadrotor_train_model_z_all.py
```

Training and test statistics of different models used for fitting the sensorimotor data. Except for one outlier all models perform on the same order of magnitude when tested on the plain sensorimotor data.

Table C.1.: Z-Axis training and test measures.

| **modeltype** | **modelsize** | **traintime fwd/inv** | **rmse (fwd)** | **rmse (inv)** |
|---|---|---|---|---|
| lin | - | 0.002051 / 0.000451 | 0.00221275 0.04719524 | 0.00819768 |
| ridge/npmdn | 100/10 | 0.000421 / 477.9887 | 0.0022272 0.04719558 | 0.00933788 |
| ogerres | 100 | 0.246541 / 0.231859 | 0.09765056 0.04949662 | 0.00828218 |
| ridge | - | 0.000662 / 0.000415 | 0.0022272 0.04719558 | 0.00819459 |
| skgp | - | 1.709583 / 2.134211 | 0.08214266 0.24205049 | 0.00699766 |
| skknn | 100/10 | 0.000922 / 0.000867 | 0.14622075 0.06257399 | 0.00697306 |
| sknnmlp | 100 | 3.684661 / 4.481415 | 0.03593596 0.04824146 | 0.00800265 |
| tnrnn (cw) | 100 | 397.8887 / 288.7724 | 0.20405665 0.06055313 | 0.00934535 |

2. Closing the loop

Closing the loop by inserting the inverse model (IM) into the sensorimotor loop yields a slightly different picture though. In the closed-loop testing scenario we generate a random point the robot should go to and use the mean squared error between this goal point and the robots actual position as the measure. In closed loop, the instantaneous sensor values

are given as input to the IM and its output is used as the raw motor signal without further processing. The goal state, which is needed as part of IM's input is computed by setting the position component to the desired position and the velocity component to the difference between current position and desired position.

Closed loop testing with Gaussian sensor noise ($\mu = 0$, $\sigma = 0.1$)

| Modeltype | Closed-loop goal error |
|---|---|
| err (lin) | 0.436919131556 |
| err (ridge) | 0.43553166434 |
| err (sknnmlp) | 0.483150503845 |
| err (skknn) | 1.69612453685 |
| err (skgp) | 5.1288925176 |
| err (ogerres) | 0.4896821421 |
| err (npmdn) | 1.24159981031 |
| err (tnrnn) | 2.20284023354 |

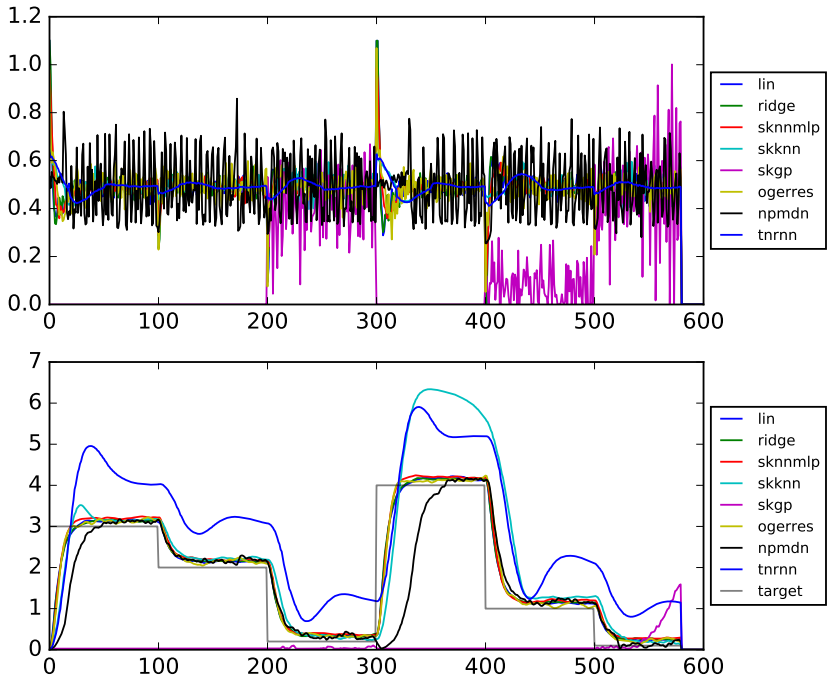



Figure C.11.: Upper panel motor output, lower panel altitude target and actual altitude under model control with sensor noise.

Closed loop testing with forward model search: A variant of implementing an inverse model is to generate an ensemble of forward simulations for random commands in a given state,

observing the distance of the predicted state from the desired state and finally executing the command with the most favourable prediction. Doing that in the 1 DoF case with 30 random commands yields a closed-loop performance comparable to that of the direct inverse model.

| Modeltype | Closed-loop goal error |
|---|---|
| err (lin) | 0.530695937297 |
| err (ridge) | 0.541257702906 |
| err (skknn) | 0.62620166885 |
| err (sknnmlp) | 0.621636823406 |
| err (skgp) | 0.578509795018 |
| err (npmdn) | 0.543719749807 |
| err (ogerres) | 0.701293737462 |
| err (tnrnn) | - |

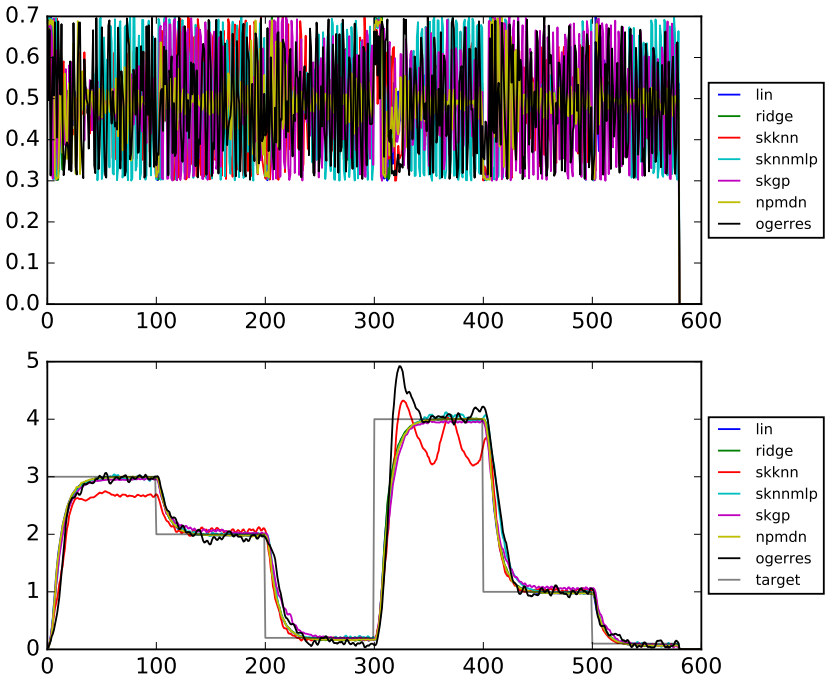



Figure C.12.: Upper panel motor output, lower panel altitude target and actual altitude under model control using forward model search.

```
# need to change variable evalfiles in im_quadrotor_plot.py for inv, inv+sigma, fwd
python im_quadrotor_plot.py --mode plot_motorbabbling_attitude_z_model_compare
```

Closed loop testing with reexploration based models

| Modeltype | Closed-loop goal error |
|---|---|
| err (lin) | 0.452104526549 |
| err (ridge) | 0.456383904481 |
| err (sknnmlp) | 0.416402269295 |
| err (skknn) | 19195.9210425 |
| err (skgp) | 5.32098186693 |
| err (ogerres) | 0.465794505839 |
| err (npmdn) | 0.457440641834 |
| err (tnrnn) | 0.905812691518 |

```
python im_quadrotor_plot.py --mode plot_motorbabbling_attitude_z_model_compare
```

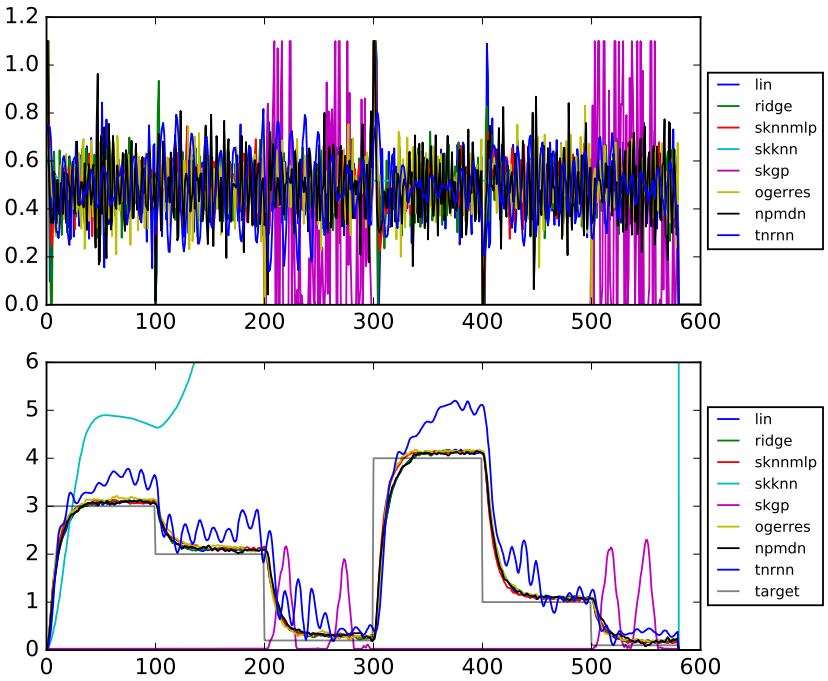



Figure C.13.: Upper panel motor output, lower panel altitude target and actual altitude under model control using direct inverse model trained on additional exploration data while under model control from motorbabbling. Sensor noise $\mu = 0$, $\sigma$ 0.1.

### 3-DoF motion

Now we extend the 1-DoF case to two additional DoF. Here, the robot orientation is held fixed so that the angular effect of the roll and pitch commands will remain constant. The exploration routine is modified according to the following scheme. The thrust command is computed via an inverse model as described the last section and the roll and pitch commands are drawn independently from a normal distribution with $\sigma = 0.1$. Thus the robot is hovering at varying altitudes while freely floating around in the x,y-plane. The dependence of the x and y acceleration on the pitch and roll command is shown in Figure C.14 for temporal shifts of one, two and three timesteps. Due to the details of lateral actuation, the strongest response to roll / pitch commands is observed after two timesteps. This stems from the fact that the underlying attitude controller needs finite time to respond to a set angle. The prediction MSE on the test set is shown in Figure C.15 for the x,y and z components separately. The closed-loop evaluation is summarized in Figure C.17 and Figure C.16.

```
# explore x,y space using z stabilization
python im_experiment.py --conf conf/im_quadrotor_exploit_z_explore_xy.py
```

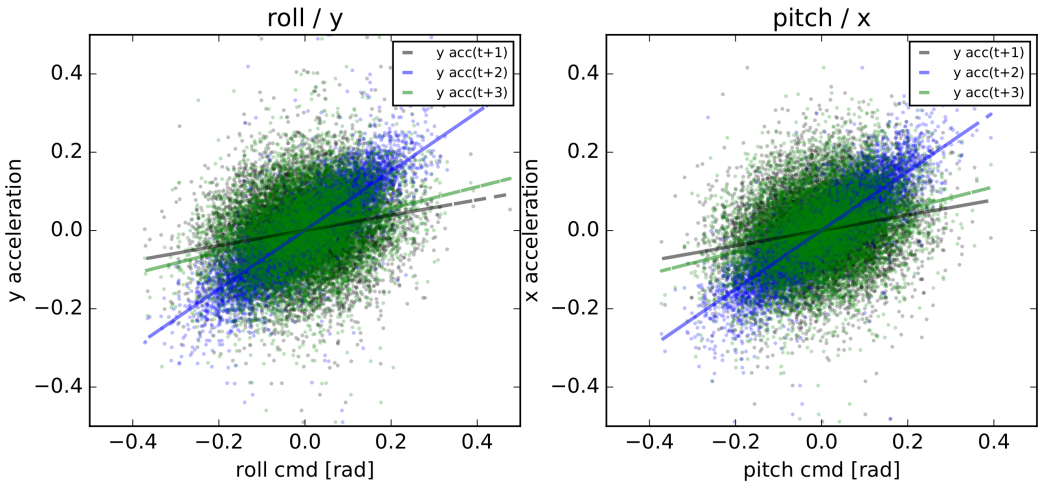



Figure C.14.: x and y acceleration response over pitch and roll commands respectively again reveals a linear relationship. The full response is distributed over several timesteps with a peak at $\tau = 2$.

### 4-DoF motion

The last degree of freedom we wish to control is the robots orientation on the z-axis, its heading. If the orientation changes, the roll/pitch commands necessary for reaching a point in the world frame of reference change as a function of the orientation angle $\psi$. Thus the model now needs to approximate a nonlinear function of the angle.
The exploration scenario is the same is in the x,y,z case, that is, during exploration the robot uses a pretrained model for controlling the altitude while both roll and pitch as well as the yaw

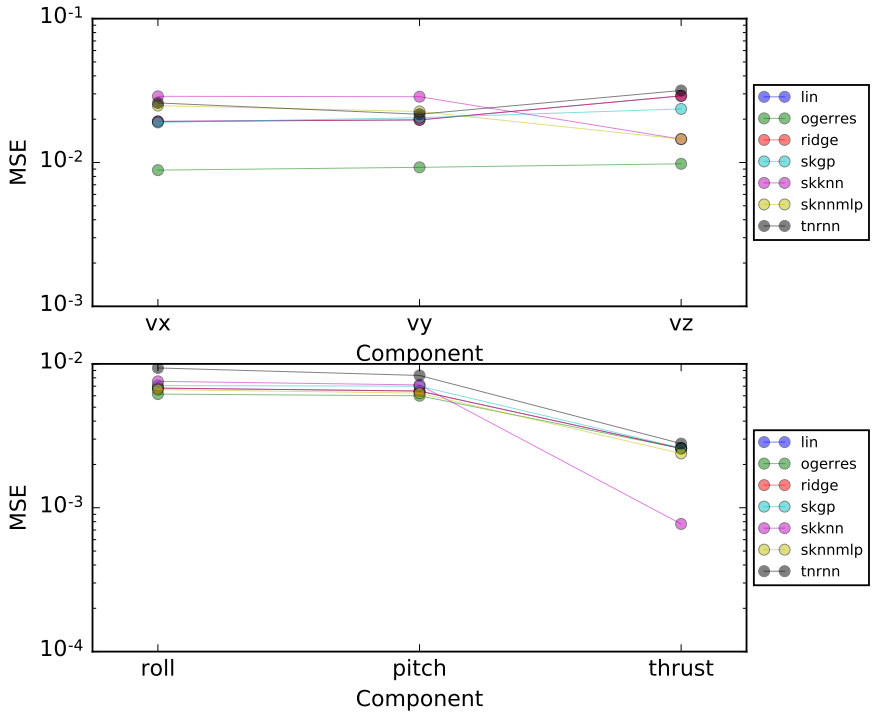



Figure C.15.: Separate MSEs on the test set for all prediction variables (components) for all model types for forward and inverse models.

commands are drawn from normal distributions. For roll/pitch we use $\sigma_{\text{lat}} = 0.1$ while for yaw we use $\sigma_{\text{yaw}} = 1.0$. The roll and pitch commands effect an angle in radian while the yaw command effects an angular rate in radian/s.

Looking at the x/y acceleration reponse to roll/pitch commands in the data emphasizes the fact that the derotation according to current yaw angle needs to be performed by the model in order the generate correct outputs, see Figure C.18. In this case we convert the yaw angle into cartesian representation as an additional preprocessing step in order to avoid the jump from -$\pi$ to $\pi$. We train the different models as for the previous cases and find comparable performance on the test set for model types in Figure C.19.

For the closed-loop evaluation we proceed again according to the schema of the previous sections with some variations. In general we observe more obvious failures of closed-loop control which can be attributed to a) model insufficiency and b) goal insufficiency. By a) we refer to the linear models which are just plain which are obviously not able to map the nonlinear rotations. By b) we mean goal configurations for which the model under consideration simply breaks down and does not give a reasonable answer.

We actually perform three different closed-loop tests. For the first one, we test all eight models without sensor noise on an *easy* goal sequence. This test already reveals the inusfficiency of some


Figure C.16.: Position vs. target MSEs for all models on the closed loop evaluation task.

of the models which are then excluded from further evaluations. The second test just repeats the easy goal sequence with sensory noise. The third and last test presents the model with a *hard* goal sequence which contains large jumps in the desired states as compared to the current state and in particular includes goals which require the robot to rotate by 180 degrees. In this situation, a unimodal model cannot deal with the ambiguity of having to equally valid motor options of reaching the state and leads to failure by *undecidedness*.

```
# explore xy and psi while controlling z
python im_experiment.py --conf conf/im_quadrotor_gs_it_xyz_re_psi.py
```

The goal to sensor information transfer across the model and environment is measured for different realizations and architectures in Figure C.22 and Figure C.23.

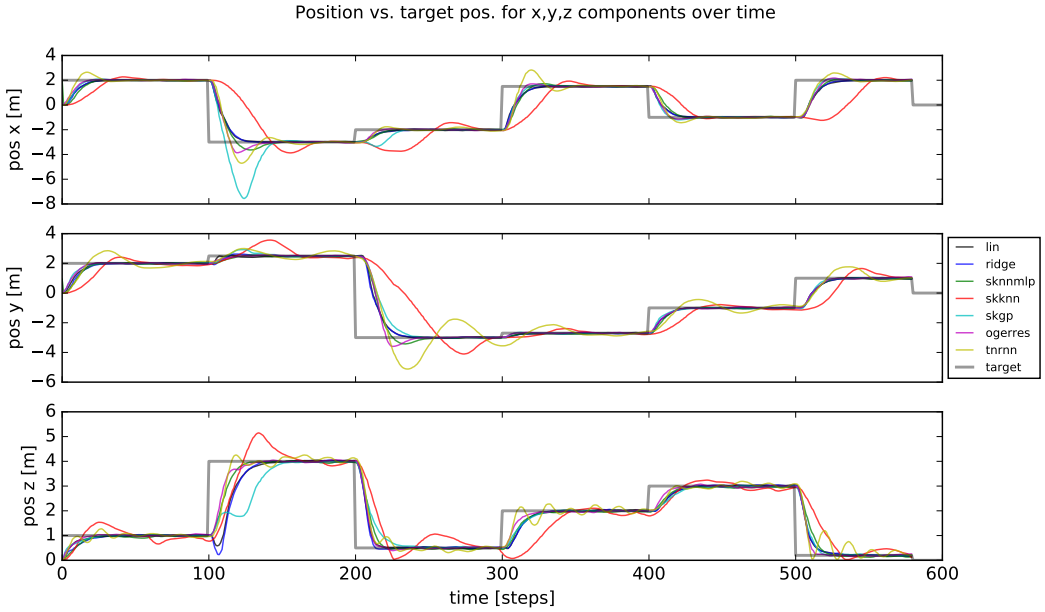



Figure C.17.: Trajectories of closed loop evaluation of all models and setpoint target.

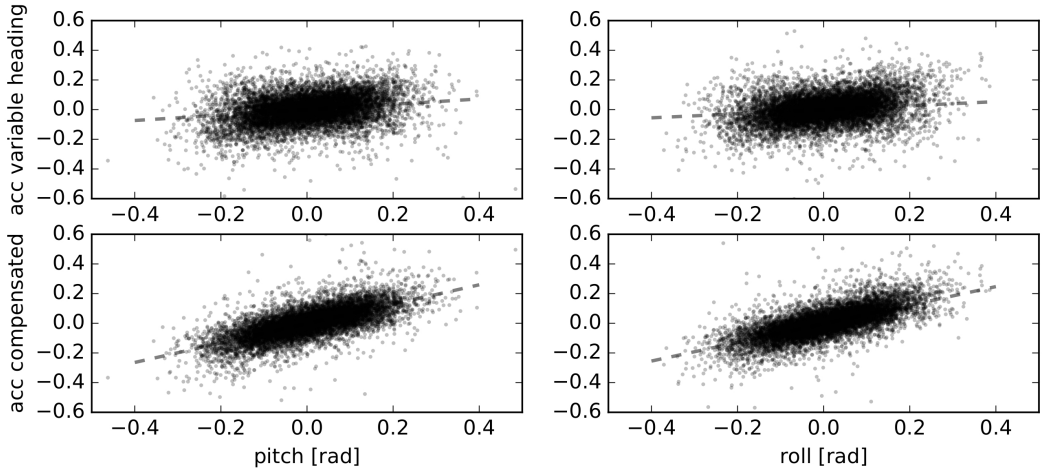



Figure C.18.: x and y acceleration response over pitch and roll commands in data resulting from arbitrary yaw angle. Not compensating for the rotation destroy the relationship.

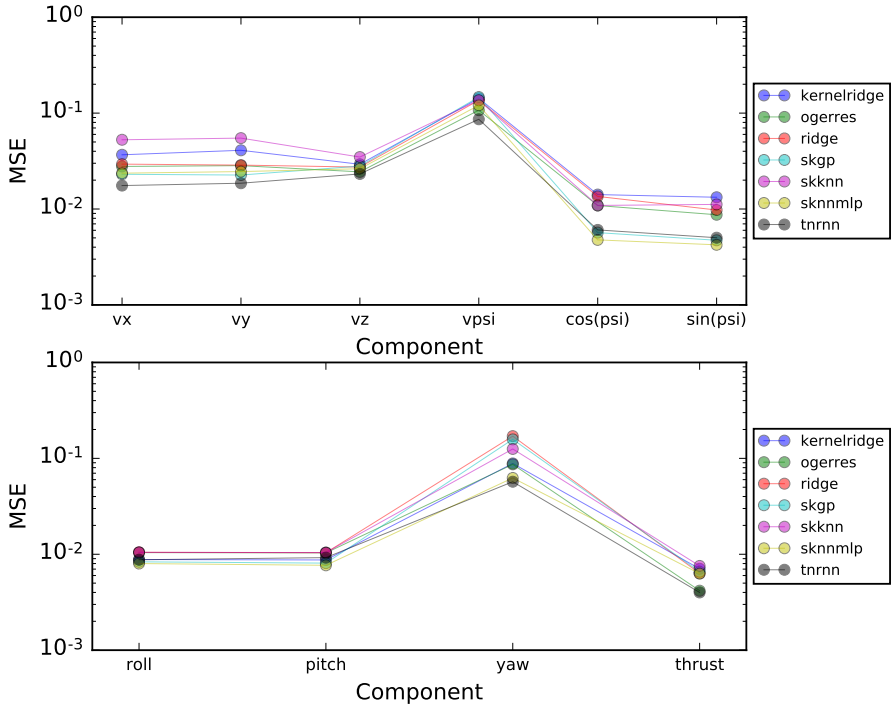



Figure C.19.: Separate MSEs on the test set for all prediction variables (components) for all model types for forward and inverse models.

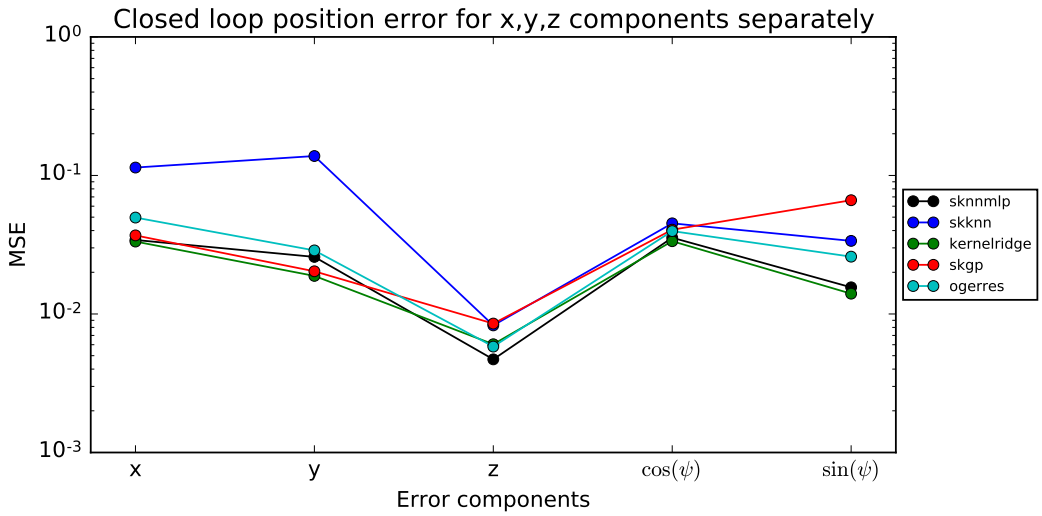



Figure C.20.: Position vs. target MSEs for all models on the closed loop evaluation task .

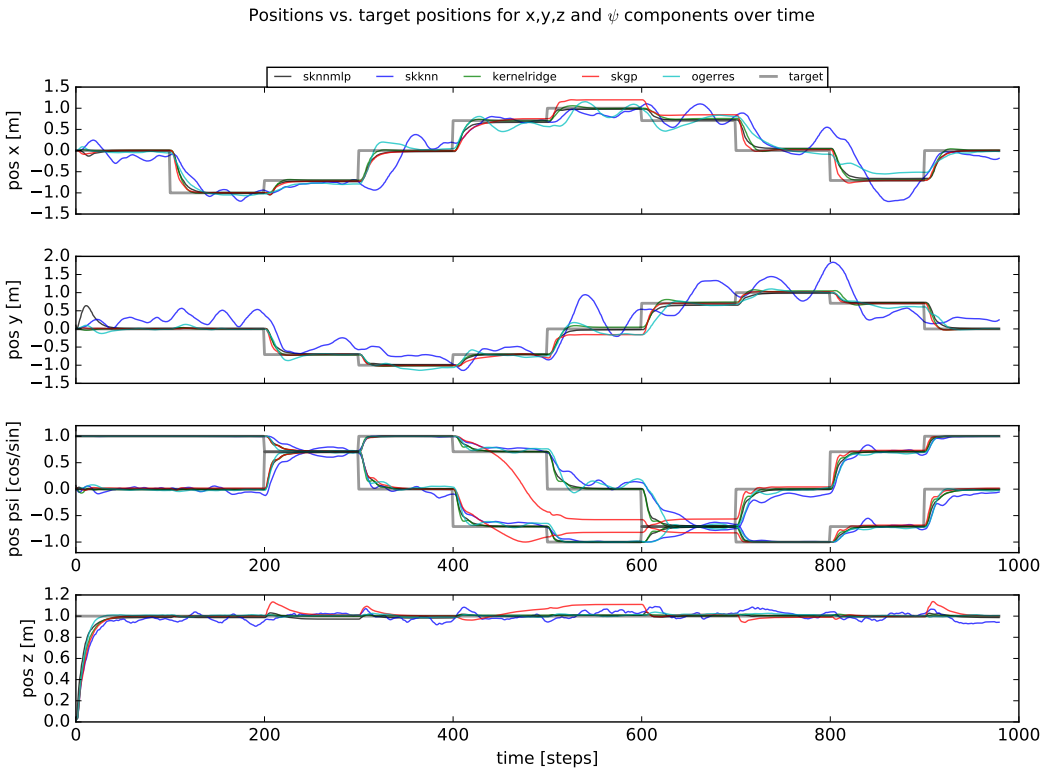



Figure C.21.: Trajectories of closed loop evaluation of four models and setpoint target.

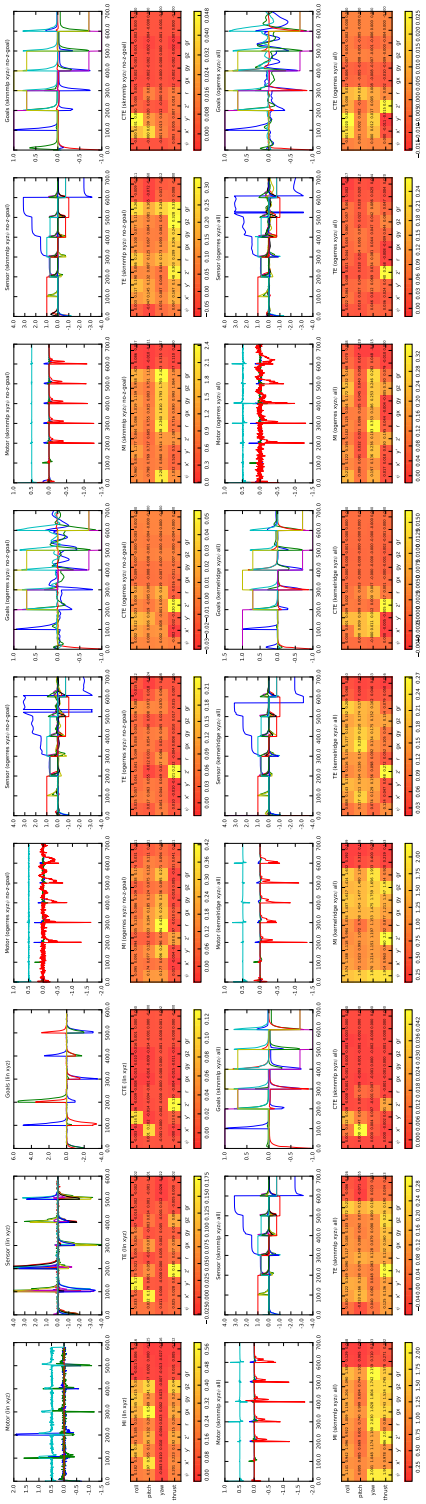


Figure C.22.: The goal to sensor information flow computed via the MI, TE, and CTE for closed-loop evaluation episodes for different trained models (Linear regression, MLP, Kernel regression, reservoir) and conditions (xyz or xyz and $\phi$). This figure has to be taken on a qualitative level indicating that the information flow signature is indeed different for these conditions.

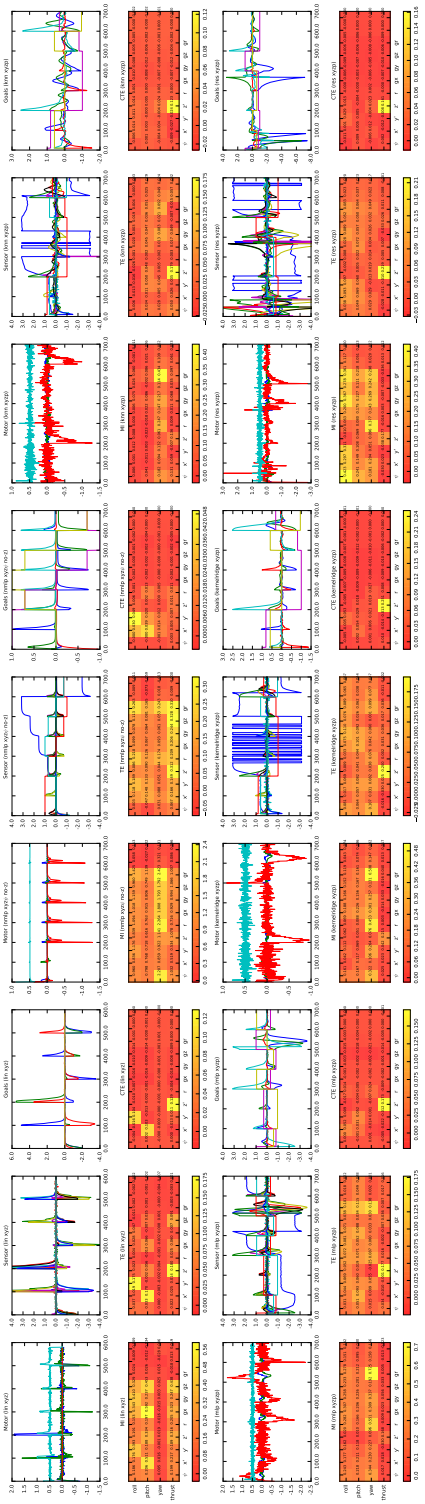


Figure C.23.: The goal to sensor information flow computed via the MI, TE, and CTE for closed-loop evaluation episodes for different trained models (Linear regression, MLP, Kernel regression, reservoir) and conditions (xyz or xyz and $\phi$). This figure has to be taken on a qualitative level indicating that the information flow signature is indeed different for these conditions.